The Power of Programs over Monoids in J

Nathan Grosshans

The Power of Programs over Monoids in **J**

Nathan Grosshans*†

Abstract

The model of programs over (finite) monoids, introduced by Barrington and Thérien, gives an interesting way to characterise the circuit complexity class NC^1 and its subclasses and showcases deep connections with algebraic automata theory. In this article, we investigate the computational power of programs over monoids in **J**, a small variety of finite aperiodic monoids. First, we give a fine hierarchy within the class of languages recognised by programs over monoids from **J**, based on the length of programs but also some parametrisation of **J**. Second, and most importantly, we make progress in understanding what regular languages can be recognised by programs over monoids in **J**. We show that those programs actually can recognise all languages from a class of restricted dot-depth one languages, using a non-trivial trick, and conjecture that this class suffices to characterise the regular languages recognised by programs over monoids in **J**.

1 Introduction

In computational complexity theory, many hard still open questions concern relationships between complexity classes that are expected to be quite small in comparison to the mainstream complexity class P of tractable languages. One of the smallest such classes is NC^1, the class of languages decided by Boolean circuits of polynomial length, logarithmic depth and bounded fan-in, a relevant and meaningful class, that has many characterisations but whose internal structure still mostly is a mystery. Indeed, among its most important subclasses, we count AC^0, CC^0 and ACC^0: all of them are conjectured to be different from each other and strictly within NC^1, but despite many efforts for several decades, this could only be proved for the first of those classes.

In the late eighties, Barrington and Thérien [3], building on Barrington's celebrated theorem [2], gave an interesting viewpoint on those conjectures, relying on algebraic automata theory. They defined the notion of a program over a monoid M: a sequence of instructions (i, f), associating through function f some element of M to the letter at position i in the input of fixed length. In that way, the program outputs an element of M for every input word, by multiplying out the elements given by the instructions for that word; acceptance or rejection then depends on that outputted element. A language of words of arbitrary length is consequently recognised in a non-uniform fashion, by a sequence of

*DI ENS, ENS, CNRS, PSL University, Paris, France, nathan.grosshans@polytechnique.edu, https://www.di.ens.fr/~ngrosshans/.

†Inria, Paris, France.

programs over some fixed monoid, one for each possible input length; when that sequence is of polynomial length, it is said that the monoid p-recognises that language. Barrington and Thérien's discovery is that NC^1 and almost all of its significant subclasses can each be exactly characterised by p-recognition over monoids taken from some suitably chosen variety of finite monoids (a class of finite monoids closed under basic operations on monoids). For instance, NC^1, AC^0, CC^0 and ACC^0 correspond exactly to p-recognition by, respectively, finite monoids, finite aperiodic monoids, finite solvable groups and finite solvable monoids. Understanding the internal structure of NC^1 thus becomes a matter of understanding what finite monoids from some particular variety are able to p-recognise.

It soon became clear that regular languages play a central role in understanding p-recognition: McKenzie, Péladeau and Thérien indeed observed [12] that finite monoids from a variety $\mathbf{V}$ and a variety $\mathbf{W}$ p-recognise the same languages if and only if they p-recognise the same regular languages. Otherwise stated, most conjectures about the internal structure of NC^1 can be reformulated as a statement about where one or several regular languages lie within that structure. This is why a line of previous works got interested into various notions of tameness, capturing the fact that for a given variety of finite monoids, p-recognition does not offer much more power than classical morphism-recognition when it comes to regular languages (see [13, 14, 11, 20, 21, 22, 10, 8]).

This paper is a contribution to an ongoing study of what regular languages can be p-recognised by monoids taken from "small" varieties, started with the author's Ph.D. thesis [7]. In a previous paper by the author with McKenzie and Segoufin [8], a novel notion of tameness was introduced and shown for the "small" variety of finite aperiodic monoids $\mathbf{DA}$. This allowed them to characterise the class of regular languages p-recognised by monoids from $\mathbf{DA}$ as those recognised by so called quasi-$\mathbf{DA}$ morphisms and represented a first small step towards a new proof that the variety $\mathbf{A}$ of finite aperiodic monoids is tame. This is a statement equivalent to Furst's, Saxe's, Sipser's [6] and Ajtai's [1] well-known lower bound result about AC^0. In [8], the authors also observed that, while $\mathbf{DA}$ "behaves well" with respect to p-recognition of regular languages, the variety $\mathbf{J}$, a subclass of $\mathbf{DA}$, does, in contrast, "behave badly" in the sense that monoids from $\mathbf{J}$ do p-recognise regular languages that are not recognised by quasi-$\mathbf{J}$ morphisms.

Now, $\mathbf{J}$ is a well-studied and fundamental variety in algebraic automata theory (see, e.g., [15, 16]), corresponding through classical morphism-recognition to the class of regular languages in which membership depends on the presence or absence of a finite set of words as subwords. This paper is a contribution to the understanding of the power of programs over monoids in $\mathbf{J}$, a knowledge that certainly does not bring us closer to a new proof of the tameness of $\mathbf{A}$ (as we are dealing with a strict subvariety of $\mathbf{DA}$), but that is motivated by the importance of $\mathbf{J}$ in algebraic automata theory and the unexpected power of programs over monoids in $\mathbf{J}$. The results we present in this article are twofold: first, we exhibit a fine hierarchy within the class of languages p-recognised by monoids from $\mathbf{J}$, depending on the length of those programs and on a parametrisation of $\mathbf{J}$; second, we show that a whole class of regular languages, that form a subclass of dot-depth one languages [16], are p-recognised by monoids from $\mathbf{J}$ while, in general, they are not recognised by any quasi-$\mathbf{J}$ morphism. This class roughly corresponds to dot-depth one languages where detection of a given factor

does work only when it does not appear too often as a subword. We actually even conjecture that this class of languages with additional positional modular counting (that is, letters can be differentiated according to their position modulo some fixed number) corresponds exactly to all those p-recognised by monoids in $\mathbf{J}$, a statement that is interesting in itself for algebraic automata theory.

Organisation of the paper. Following the present introduction, Section 2 is dedicated to the necessary preliminaries. In Section 3, we present the results about the fine hierarchy and in Section 4 we expose the results concerning the regular languages p-recognised by monoids from $\mathbf{J}$. Section 5 gives a short conclusion.

Note. This article is based on unpublished parts of the author's Ph.D. thesis [7].

2 Preliminaries

2.1 Various mathematical materials

We assume the reader is familiar with the basics of formal language theory, semigroup theory and recognition by morphisms, that we might designate by classical recognition; for those, we only specify some things and refer the reader to the two classical references of the domain by Eilenberg [4, 5] and Pin [15].

General notations and conventions. Let $i, j \in \mathbb{N}$. We shall denote by $[\![i, j]\!]$ the set of all $n \in \mathbb{N}$ verifying $i \leq n \leq j$. We shall also denote by $[i]$ the set $[\![1, i]\!]$. Given some set E, we shall denote by $\mathfrak{P}(E)$ the powerset of E. All our alphabets and words will always be finite; the empty word will be denoted by ε.

Varieties and languages. A *variety of monoids* is a class of finite monoids closed under submonoids, Cartesian product and morphic images. A *variety of semigroups* is defined similarly. When dealing with varieties, we consider only finite monoids and semigroups, each having an *idempotent power*, a smallest $\omega \in \mathbb{N}_{>0}$ such that $x^\omega = x^{2\omega}$ for any element x. To give an example, the variety of finite aperiodic monoids, denoted by $\mathbf{A}$, contains all finite monoids M such that, given ω its idempotent power, $x^\omega = x^{\omega+1}$ for all $x \in M$.

To each variety $\mathbf{V}$ of monoids or semigroups we associate the class $\mathcal{L}(\mathbf{V})$ of languages such that, respectively, their syntactic monoid or semigroup belongs to $\mathbf{V}$. For instance, $\mathcal{L}(\mathbf{A})$ is well-known to be the class of star-free languages.

Quasi V languages. If S is a semigroup we denote by S^1 the monoid S if S is already a monoid and $S \cup \{1\}$ otherwise.

The following definitions are taken from [17]. Let φ be a surjective morphism from Σ^* to a finite monoid M. For all k consider the subset $\varphi(\Sigma^k)$ of M (where Σ^k is the set of words over Σ of length k). As M is finite there is a k such that $\varphi(\Sigma^{2k}) = \varphi(\Sigma^k)$. This implies that $\varphi(\Sigma^k)$ is a semigroup. The semigroup given by the smallest such k is called the *stable semigroup of* φ. If S is the stable semigroup of φ, S^1 is called *the stable monoid of* φ. If $\mathbf{V}$ is a variety of monoids or semigroups, then we shall denote by $\mathbf{QV}$ the class of such surjective

morphisms whose stable monoid or semigroup, respectively, is in $\mathbf{V}$ and by $\mathcal{L}(\mathbf{QV})$ the class of languages whose syntactic morphism is in $\mathbf{QV}$.

Programs over monoids. Programs over monoids form a non-uniform model of computation, first defined by Barrington and Thérien [3], extending Barrington's permutation branching program model [2]. Let M be a finite monoid and Σ an alphabet. A *program P over M on Σ^n* is a finite sequence of instructions of the form (i, f) where $i \in [n]$ and $f \in M^\Sigma$; said otherwise, it is a word over $([n] \times M^\Sigma)$. The *length* of P, denoted by $|P|$, is the number of its instructions. The program P defines a function from Σ^n to M as follows. On input $w \in \Sigma^n$, each instruction (i, f) outputs the monoid element $f(w_i)$. A sequence of instructions then yields a sequence of elements of M and their product is the output $P(w)$ of the program. A language $L \subseteq \Sigma^n$ is consequently recognised by P whenever there exists $F \subseteq M$ such that $L = P^{-1}(F)$.

A language L over Σ is *recognised* by a sequence of programs $(P_n)_{n\in\mathbb{N}}$ over some finite monoid M if for each n, the program P_n is on Σ^n and recognises $L^{=n} = L \cap \Sigma^n$. We say $(P_n)_{n\in\mathbb{N}}$ is of length $s(n)$ for $s\colon \mathbb{N} \to \mathbb{N}$ whenever $|P_n| = s(n)$ for all $n \in \mathbb{N}$ and that it is of length at most $s(n)$ whenever there exists $\alpha \in \mathbb{R}_{>0}$ verifying $|P_n| \leq \alpha \cdot s(n)$ for all $n \in \mathbb{N}$.

For $s\colon \mathbb{N} \to \mathbb{N}$ and $\mathbf{V}$ a variety of monoids, we denote by $\mathcal{P}(\mathbf{V}, s(n))$ the class of languages recognised by sequences of programs over monoids in $\mathbf{V}$ of length at most $s(n)$. The class $\mathcal{P}(\mathbf{V}) = \bigcup_{k\in\mathbb{N}} \mathcal{P}\big(\mathbf{V}, n^k\big)$ is then the class of languages p-recognised by a monoid in $\mathbf{V}$, i.e. recognised by sequences of programs over monoids in $\mathbf{V}$ of polynomial length.

The following is an important property of $\mathcal{P}(\mathbf{V})$.

Proposition 2.1 ([12, Corollary 3.5])**.** *Let $\mathbf{V}$ be a variety of monoids, then $\mathcal{P}(\mathbf{V})$ is closed under Boolean operations.*

Given two alphabets Σ and Γ, a Γ-program on Σ^n for $n \in \mathbb{N}$ is defined just like a program over some finite monoid M on Σ^n, except that instructions output letters from Γ and thus that the program outputs words over Γ. Let now $L \subseteq \Sigma^*$ and $K \subseteq \Gamma^*$. We say that L *program-reduces to* K if and only if there exists a sequence $(\Psi_n)_{n\in\mathbb{N}}$ of Γ-programs (the program-reduction) such that Ψ_n is on Σ^n and $L^{=n} = \Psi_n^{-1}(K^{=|\Psi_n|})$ for each $n \in \mathbb{N}$. The following proposition shows closure of $\mathcal{P}(\mathbf{V})$ also under program-reductions.

Proposition 2.2 ([7, Proposition 3.3.12 and Corollary 3.4.3])**.** *Let Σ and Γ be two alphabets. Let $\mathbf{V}$ be a variety of monoids. Given $K \subseteq \Gamma^*$ in $\mathcal{P}(\mathbf{V}, s(n))$ for $s\colon \mathbb{N} \to \mathbb{N}$ and $L \subseteq \Sigma^*$ from which there exists a program-reduction to K of length $t(n)$, for $t\colon \mathbb{N} \to \mathbb{N}$, we have that $L \in \mathcal{P}(\mathbf{V}, s(t(n)))$. In particular, when K is recognised (classically) by a monoid in $\mathbf{V}$, we have that $L \in \mathcal{P}(\mathbf{V}, t(n))$.*

2.2 Tameness and the variety J

We won't introduce any of the proposed notions of tameness but will only state that the main consequence for a variety of monoids $\mathbf{V}$ to be tame in the sense of [8] is that $\mathcal{P}(\mathbf{V}) \cap \mathcal{R}\mathsf{eg} \subseteq \mathcal{L}(\mathbf{QV})$. This consequence has far-reaching implications from a computational-complexity-theoretic standpoint when $\mathcal{P}(\mathbf{V})$ happens to be equal to a circuit complexity class. For instance, tameness for $\mathbf{A}$ implies that $\mathcal{P}(\mathbf{A}) \cap \mathcal{R}\mathsf{eg} \subseteq \mathcal{L}(\mathbf{QA})$, which is equivalent to the fact that AC^0

does not contain the language MOD_m of words over $\{0,1\}$ containing a number of 1s not divisible by m for any $m \in \mathbb{N}, m \geq 2$ (a central result in complexity theory [6, 1]).

Let us now define the variety of monoids $\mathbf{J}$. A finite monoid M of idempotent power ω belongs to $\mathbf{J}$ if and only if $(xy)^\omega = (xy)^\omega x = y(xy)^\omega$ for all $x, y \in M$. It is a strict subvariety of the variety $\mathbf{DA}$, containing all finite monoids M of idempotent power ω such that $(xy)^\omega = (xy)^\omega x (xy)^\omega$ for all $x, y \in M$, itself a strict subvariety of $\mathbf{A}$. The variety $\mathbf{J}$ is a "small" one, well within $\mathbf{A}$.

We now give some specific definitions and results about $\mathbf{J}$ that we will use, based essentially on [9], but also on [15, Chapter 4, Section 1].

For some alphabet Σ and each $k \in \mathbb{N}$, let us define the equivalence relation $\sim_k$ on Σ^* by $u \sim_k v$ if and only if u and v have the same set of k-subwords (subwords of length at most k), for all $u, v \in \Sigma^*$. The relation $\sim_k$ is a congruence of finite index on Σ^*. For an alphabet Σ and a word $u \in \Sigma^*$, we shall write $u \sqcup\!\sqcup \Sigma^*$ for the language of all words over Σ having u as a subword. In the following, we consider that $\sqcup\!\sqcup$ has precedence over $\cup$ and $\cap$ (but of course not over concatenation).

We define the *class of piecewise testable languages* $\mathcal{PT}$ as the class of regular languages such that for every alphabet Σ, we associate to Σ^* the set $\mathcal{PT}(\Sigma^*)$ of all languages over Σ that are Boolean combinations of languages of the form $u \sqcup\!\sqcup \Sigma^*$ where $u \in \Sigma^*$. In fact, $\mathcal{PT}(\Sigma^*)$ is the set of languages over Σ equal to a union of $\sim_k$-classes for some $k \in \mathbb{N}$ (see [18]). Simon showed [18] that a language is piecewise testable if and only if its syntactic monoid is in $\mathbf{J}$, i.e. $\mathcal{PT} = \mathcal{L}(\mathbf{J})$.

We can define a hierarchy of piecewise testable languages in a natural way. For $k \in \mathbb{N}$, let the *class of k-piecewise testable languages* $\mathcal{PT}_k$ be the class of regular languages such that for every alphabet Σ, we associate to Σ^* the set $\mathcal{PT}_k(\Sigma^*)$ of all languages over Σ that are Boolean combinations of languages of the form $u \sqcup\!\sqcup \Sigma^*$ where $u \in \Sigma^*$ with $|u| \leq k$. We then have that $\mathcal{PT}_k(\Sigma^*)$ is the set of languages over Σ equal to a union of $\sim_k$-classes. Let us define $\mathbf{J_k}$ the inclusion-wise smallest variety of monoids containing the quotients of Σ^* by $\sim_k$ for any alphabet Σ: we have that a language is k-piecewise testable if and only if its syntactic monoid belongs to $\mathbf{J_k}$, i.e. $\mathcal{PT}_k = \mathcal{L}(\mathbf{J_k})$. (See [9, Section 3].)

3 Fine Hierarchy

The first part of our investigation of the computational power of programs over monoids in $\mathbf{J}$ concerns the influence of the length of programs on their computational capabilities.

We say two programs over a same monoid on the same set of input words are *equivalent* if and only if they recognise the same languages. Tesson and Thérien proved in [23] that for any monoid M in $\mathbf{DA}$, there exists some $k \in \mathbb{N}$ such that for any alphabet Σ there is a constant $c \in \mathbb{N}_{>0}$ verifying that any program over M on Σ^n for $n \in \mathbb{N}$ is equivalent to a program over M on Σ^n of length at most $c \cdot n^k$. Since $\mathbf{J} \subset \mathbf{DA}$, any monoid in $\mathbf{J}$ does also have this property. However, this does not imply that there exists some $k \in \mathbb{N}$ working for all monoids in $\mathbf{J}$, i.e. that $\mathcal{P}(\mathbf{J})$ collapses to $\mathcal{P}\big(\mathbf{J}, n^k\big)$.

In this section, we show on the one hand that, as for $\mathbf{DA}$, while $\mathcal{P}(\mathbf{J}, s(n))$ collapses to $\mathcal{P}(\mathbf{J})$ for any super-polynomial function $s\colon \mathbb{N} \to \mathbb{N}$, there does not

exist any $k \in \mathbb{N}$ such that $\mathcal{P}(\mathbf{J})$ collapses to $\mathcal{P}\big(\mathbf{J}, n^k\big)$; and on the other hand that $\mathcal{P}(\mathbf{J_k})$ does optimally collapse to $\mathcal{P}\big(\mathbf{J_k}, n^{\lceil k/2 \rceil}\big)$ for each $k \in \mathbb{N}$.

3.1 Strict hierarchy

Given $k, n \in \mathbb{N}$, we say that σ is a *k-selector over n* if σ is a function of $\mathfrak{P}([n])^{[n]^k}$ that associates a subset of $[n]$ to each vector in $[n]^k$. For any sequence $\Delta = (\sigma_n)_{n\in\mathbb{N}}$ such that σ_n is a k-selector over n for each $n \in \mathbb{N}$ — a sequence we will call a *sequence of k-selectors* —, we set $L_\Delta = \bigcup_{n\in\mathbb{N}} K_{n,\sigma_n}$, where for each $n \in \mathbb{N}$, the language K_{n,σ_n} is the set of words over $\{0,1\}$ of length $(k+1)\cdot n$ that can be decomposed into $k+1$ consecutive blocks $u^{(1)}, u^{(2)}, \ldots, u^{(k)}, v$ of n letters where the first k blocks each contain 1 exactly once and uniquely define a vector ρ in $[n]^k$, where for all $i \in [k]$, ρ_i is given by the position of the only 1 in $u^{(i)}$ (i.e. $u^{(i)}_{\rho_i} = 1$) and v is such that there exists $j \in \sigma_n(\rho)$ verifying that v_j is 1. Observe that for any k-selector σ_0 over 0, we have $K_{0,\sigma_0} = \emptyset$.

We now proceed similarly to what has been done in Subsection 5.1 in [8] to show, on one hand, that for all $k \in \mathbb{N}$, there is a monoid M_k in $\mathbf{J_{2k+1}}$ such that for any sequence of k-selectors Δ, the language L_Δ is recognised by a sequence of programs over M_k of length at most n^{k+1}; and, on the other hand, that for all $k \in \mathbb{N}$ there is a sequence of k-selectors Δ such that for any finite monoid M and any sequence of programs $(P_n)_{n\in\mathbb{N}}$ over M of length at most n^k, the language L_Δ is not recognised by $(P_n)_{n\in\mathbb{N}}$.

We obtain the following proposition; the proof can be found in Appendix A.

Proposition 3.1. *For all $k \in \mathbb{N}$, we have $\mathcal{P}\big(\mathbf{J}, n^k\big) \subset \mathcal{P}\big(\mathbf{J}, n^{k+1}\big)$. More precisely, for all $k \in \mathbb{N}$ and $d \in \mathbb{N}, d \leq \lceil \frac{k}{2} \rceil - 1$, we have $\mathcal{P}\big(\mathbf{J_k}, n^d\big) \subset \mathcal{P}\big(\mathbf{J_k}, n^{d+1}\big)$.*

3.2 Collapse

Looking at Proposition 3.1, it looks at first glance rather strange that, for each $k \in \mathbb{N}$, we can only prove strictness of the hierarchy inside $\mathcal{P}(\mathbf{J_k})$ up to exponent $\lceil \frac{k}{2} \rceil$. We now show, in a way similar to Subsection 5.2 in [8], that in fact $\mathcal{P}(\mathbf{J_k})$ does collapse to $\mathcal{P}\big(\mathbf{J_k}, n^{\lceil k/2 \rceil}\big)$ for all $k \in \mathbb{N}$, showing Proposition 3.1 to be optimal in some sense. Due to space constraints, we leave the proof to Appendix A.

Proposition 3.2. *Let $k \in \mathbb{N}$. Let $M \in \mathbf{J_k}$ and Σ be an alphabet. Then there exists a constant $c \in \mathbb{N}_{>0}$ such that any program over M on Σ^n for $n \in \mathbb{N}$ is equivalent to a program over M on Σ^n of length at most $c \cdot n^{\lceil k/2 \rceil}$.*

In particular, $\mathcal{P}(\mathbf{J_k}) = \mathcal{P}\big(\mathbf{J_k}, n^{\lceil k/2 \rceil}\big)$ for all $k \in \mathbb{N}$.

4 Regular Languages in $\mathcal{P}(\mathbf{J})$

The second part of our investigation of the computational power of programs over monoids in $\mathbf{J}$ is dedicated to understanding exactly what regular languages can be p-recognised by monoids in $\mathbf{J}$.

4.1 Non-tameness of J

It is shown in [8] that $\mathcal{P}(\mathbf{J}) \cap \mathcal{R}\mathsf{eg} \not\subseteq \mathcal{L}(\mathbf{QJ})$, thus giving an example of a well-known subvariety of $\mathbf{A}$ for which p-recognition allows to do unexpected things when recognising a regular language. How far does this unexpected power go?

The first thing to notice is that, though none of them is in $\mathcal{L}(\mathbf{QJ})$, all languages of the form $\Sigma^* u$ and $u\Sigma^*$ for Σ an alphabet and $u \in \Sigma^+$ are in $\mathcal{P}(\mathbf{J})$. Indeed, each of them can be recognised by a sequence of constant-length programs over the syntactic monoid of $u \sqcup\!\sqcup \Sigma^*$: for every input length, just output the image, through the syntactic morphism of $u \sqcup\!\sqcup \Sigma^*$, of the word made of the $|u|$ first or last letters. So, informally stated, programs over monoids in $\mathbf{J}$ can check for some constant-length beginning or ending of their input words.

But they can do much more. Indeed, the language $(a+b)^* ac^+$ does not belong to $\mathcal{L}(\mathbf{QJ})$ (compute the stable monoid), yet it is in $\mathcal{P}(\mathbf{J})$. The crucial insight is that it can be program-reduced in linear length to the piecewise testable language of all words over $\{a, b, c\}$ having ca as a subword but not the subwords cca, caa and cb by using the following trick (that we shall call "feedback-sweeping") for input length $n \in \mathbb{N}$: read the input letters in the order $2, 1, 3, 2, 4, 3, 5, 4, \ldots, n, n-1$, output the letters read. This has already been observed in [8, Proposition 5]; here we give a formal proof of the following lemma, to be found in Appendix B.

Lemma 4.1. $(a+b)^* ac^+ \in \mathcal{P}(\mathbf{J}, n)$.

Using variants of the "feedback-sweeping" reading technique, we can prove that the phenomenon just described is not an isolated case.

Lemma 4.2. *The languages* $(a+b)^* ac^+$, $(a+b)^* ac^+ a(a+b)^*$, $c^+ a(a+b)^* ac^+$, $(a+b)^* bac^+$ *and* $(a+b)^* ac^+ (a+b)^* ac^+$ *do all belong to* $\mathcal{P}(\mathbf{J}) \setminus \mathcal{L}(\mathbf{QJ})$.

Hence, we are tempted to say that there are "much more" regular languages in $\mathcal{P}(\mathbf{J})$ than just those in $\mathcal{L}(\mathbf{QJ})$, even though it is not clear to us whether $\mathcal{L}(\mathbf{QJ}) \subseteq \mathcal{P}(\mathbf{J})$ or not. But can we show any upper bound on $\mathcal{P}(\mathbf{J}) \cap \mathcal{R}\mathsf{eg}$? It turns out that we can, relying on two known results.

First, since $\mathbf{J} \subseteq \mathbf{DA}$, we have $\mathcal{P}(\mathbf{J}) \subseteq \mathcal{P}(\mathbf{DA})$, so Theorem 6 in [8], that states $\mathcal{P}(\mathbf{DA}) \cap \mathcal{R}\mathsf{eg} = \mathcal{L}(\mathbf{QDA})$, implies that $\mathcal{P}(\mathbf{J}) \cap \mathcal{R}\mathsf{eg} \subseteq \mathcal{L}(\mathbf{QDA})$.

Second, let us define an important superclass of the class of piecewise testable languages. Let Σ be an alphabet and $u_1, \ldots, u_k \in \Sigma^+$ ($k \in \mathbb{N}_{>0}$); we define $[u_1, \ldots, u_k] = \Sigma^* u_1 \Sigma^* \cdots \Sigma^* u_k \Sigma^*$. The *class of dot-depth one languages* is the class of Boolean combinations of languages of the form $\Sigma^* u$, $u\Sigma^*$ and $[u_1, \ldots, u_k]$ for Σ an alphabet, $k \in \mathbb{N}_{>0}$ and $u, u_1, \ldots, u_k \in \Sigma^+$. The inclusion-wise smallest variety of semigroups containing all syntactic semigroups of dot-depth one languages is denoted by $\mathbf{J} * \mathbf{D}$ and verifies that $\mathcal{L}(\mathbf{J} * \mathbf{D})$ is exactly the class of dot-depth one languages. (See [19, 11, 16].) It has been shown in [11, Corollary 8] that $\mathcal{P}(\mathbf{J} * \mathbf{D}) \cap \mathcal{R}\mathsf{eg} = \mathcal{L}(\mathbf{Q}(\mathbf{J} * \mathbf{D}))$ (if we extend the program-over-monoid formalism in the obvious way to finite semigroups). Now, we have $\mathbf{J} \subseteq \mathbf{J} * \mathbf{D}$, so that $\mathcal{P}(\mathbf{J}) \subseteq \mathcal{P}(\mathbf{J} * \mathbf{D})$ and hence $\mathcal{P}(\mathbf{J}) \cap \mathcal{R}\mathsf{eg} \subseteq \mathcal{L}(\mathbf{Q}(\mathbf{J} * \mathbf{D}))$.

To summarise, we have the following.

Proposition 4.3. $\mathcal{P}(\mathbf{J}) \cap \mathcal{R}\mathsf{eg} \subseteq \mathcal{L}(\mathbf{QDA}) \cap \mathcal{L}(\mathbf{Q}(\mathbf{J} * \mathbf{D}))$.

In fact, we conjecture that the inverse inclusion does also hold.

Conjecture 1. $\mathcal{P}(\mathbf{J}) \cap \mathcal{R}\mathrm{eg} = \mathcal{L}(\mathbf{QDA}) \cap \mathcal{L}(\mathbf{Q}(\mathbf{J} * \mathbf{D}))$.

Why do we think this should be true? Though, for a given alphabet Σ, we cannot decide whether some word $u \in \Sigma^+$ of length at least 2 appears as a factor of any given word w in Σ^* with programs over monoids in $\mathbf{J}$ (because $\Sigma^* u \Sigma^* \notin \mathcal{L}(\mathbf{QDA})$), Lemma 4.2 and the possibilities offered by the "feedback-sweeping" technique give the impression that we can do it when we are guaranteed that u appears at most a fixed number of times in w, which seems somehow to be what dot-depth one languages become when restricted to belong to $\mathcal{L}(\mathbf{QDA})$. This intuition motivates the definition of *threshold dot-depth one languages*.

4.2 Threshold dot-depth one languages

The idea behind the definition of threshold dot-depth one languages is that we take the basic building blocks of dot-depth one languages, of the form $[u_1, \ldots, u_k]$ for an alphabet Σ, for $k \in \mathbb{N}_{>0}$ and $u_1, \ldots, u_k \in \Sigma^+$, and restrict them so that, given $l \in \mathbb{N}_{>0}$, membership of a word does really depend on the presence of a given word u_i as a factor if and only if it appears less than l times as a subword.

Definition 4.4. Let Σ be an alphabet. For all $u \in \Sigma^+$ and $l \in \mathbb{N}_{>0}$, we define $[u]_l$ to be the language of words over Σ containing u^l as a subword or u as a factor, i.e. $[u]_l = \Sigma^* u \Sigma^* \cup u^l \sqcup\!\sqcup \Sigma^*$. Then, for all $u_1, \ldots, u_k \in \Sigma^+$ ($k \in \mathbb{N}, k \geq 2$) and $l \in \mathbb{N}_{>0}$, we define $[u_1, \ldots, u_k]_l = [u_1]_l \cdots [u_k]_l$.

Obviously, for each Σ an alphabet, $k \in \mathbb{N}_{>0}$ and $u_1, \ldots, u_k \in \Sigma^+$, the language $[u_1, \ldots, u_k]_1$ equals $u_1 \cdots u_k \sqcup\!\sqcup \Sigma^*$. Over $\{a, b, c\}$, the language $[ab, c]_3$ contains all words containing a letter c verifying that in the prefix up to that letter, *ababab* appears as a subword or *ab* appears as a factor. Finally, the language $(a+b)^* a c^+$ over $\{a, b, c\}$ of Lemma 4.1 is equal to $[c, a]_2^{\complement} \cap [c, b]_2^{\complement} \cap [ac]_2$.

We then define a *threshold dot-depth one language* as any Boolean combination of languages of the form $\Sigma^* u$, $u\Sigma^*$ and $[u_1, \ldots, u_k]_l$ for Σ an alphabet, for $k, l \in \mathbb{N}_{>0}$ and $u, u_1, \ldots, u_k \in \Sigma^+$.

Confirming the intuition briefly given above, the technique of "feedback-sweeping" can indeed be pushed further to prove that the whole class of threshold dot-depth one languages is contained in $\mathcal{P}(\mathbf{J})$, and we dedicate the remainder of this section to prove it. Concerning Conjecture 1, our intuition leads us to believe that, in fact, the class of threshold dot-depth one languages with additional positional modular counting is exactly $\mathcal{L}(\mathbf{QDA}) \cap \mathcal{L}(\mathbf{Q}(\mathbf{J} * \mathbf{D}))$. We simply refer the interested reader to Section 5.4 of the author's Ph.D. thesis [7], that contains a partial result supporting this belief, too technical and long to be presented here.

Let us now move on to the proof of the following theorem.

Theorem 4.5. *Every threshold dot-depth one language belongs to* $\mathcal{P}(\mathbf{J})$.

As $\mathcal{P}(\mathbf{J})$ is closed under Boolean operations (Proposition 2.1), our goal is to prove, given an alphabet Σ, given $l \in \mathbb{N}_{>0}$ and $u_1, \ldots, u_k \in \Sigma^+$ ($k \in \mathbb{N}_{>0}$), that $[u_1, \ldots, u_k]_l$ is in $\mathcal{P}(\mathbf{J})$; the case of $\Sigma^* u$ and $u\Sigma^*$ for $u \in \Sigma^+$ is easily handled (see the discussion at the beginning of Subsection 4.1). To do this, we need to put $[u_1, \ldots, u_k]_l$ in some normal form. It is readily seen that $[u_1, \ldots, u_k]_l = \bigcup_{q_1, \ldots, q_k \in \{1, l\}} L^{(l)}_{(u_1, q_1)} \cdots L^{(l)}_{(u_k, q_k)}$ where the $L^{(l)}_{(u_i, q_i)}$'s are defined thereafter.

Definition 4.6. Let Σ be an alphabet.

For all $u \in \Sigma^+$, $l \in \mathbb{N}_{>0}$ and $\alpha \in [l]$, set $L^{(l)}_{(u,\alpha)} = \begin{cases} \Sigma^* u \Sigma^* & \text{if } \alpha < l \\ u^l \sqcup\!\sqcup \Sigma^* & \text{otherwise} \end{cases}$.

Building directly a sequence of programs over a monoid in $\mathbf{J}$ that decides $L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)}$ for some alphabet Σ and $q_1, \ldots, q_k \in \{1, l\}$ seems however tricky. We need to split things further by controlling precisely how many times each u_i for $i \in [k]$ appears in the right place when it does less than l times. To do this, we consider, for each $\alpha \in [l]^k$, the language $R^\alpha_l(u_1, \ldots, u_k)$ defined below.

Definition 4.7. Let Σ be an alphabet.

For all $u_1, \ldots, u_k \in \Sigma^+$ ($k \in \mathbb{N}_{>0}$), $l \in \mathbb{N}_{>0}$, $\alpha \in [l]^k$, we set

$$R^\alpha_l(u_1, \ldots, u_k) = (u_1{}^{\alpha_1} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^* \cap \bigcap_{i \in [k], \alpha_i < l} \big((u_1{}^{\alpha_1} \cdots u_i{}^{\alpha_i+1} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big)^{\complement} .$$

Now, for a given $\alpha \in [l]^k$, we are interested in the words of $R^\alpha_l(u_1, \ldots, u_k)$ such that for each $i \in [k]$ verifying $\alpha_i < l$, the word u_i indeed appears as a factor in the right place. We thus introduce a last language $S^\alpha_l(u_1, \ldots, u_k)$ defined as follows.

Definition 4.8. Let Σ be an alphabet.

For all $u_1, \ldots, u_k \in \Sigma^+$ ($k \in \mathbb{N}_{>0}$), $l \in \mathbb{N}_{>0}$, $\alpha \in [l]^k$, we set

$$S^\alpha_l(u_1, \ldots, u_k) = \bigcap_{i \in [k], \alpha_i < l} \big((u_1{}^{\alpha_1} \cdots u_{i-1}{}^{\alpha_{i-1}}) \sqcup\!\sqcup \Sigma^*\big) u_i \big((u_{i+1}{}^{\alpha_{i+1}} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big).$$

We now have the normal form we were looking for to prove Theorem 4.5: $[u_1, \ldots, u_k]_l$ is equal to the union, over all $\alpha \in [l]^k$, of the intersection of $R^\alpha_l(u_1, \ldots, u_k)$ and $S^\alpha_l(u_1, \ldots, u_k)$. Though rather intuitive, the correctness of this decomposition is not so straightforward to prove and, actually, we can only prove it when for each $i \in [k]$, the letters in u_i are all distinct. (The tedious proof is to be found in Appendix B.)

Lemma 4.9. *Let Σ be an alphabet, $l \in \mathbb{N}_{>0}$ and $u_1, \ldots, u_k \in \Sigma^+$ ($k \in \mathbb{N}_{>0}$) such that for each $i \in [k]$, the letters in u_i are all distinct. Then,*

$$\bigcup_{q_1, \ldots, q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)} = \bigcup_{\alpha \in [l]^k} \big(R^\alpha_l(u_1, \ldots, u_k) \cap S^\alpha_l(u_1, \ldots, u_k)\big) .$$

Our goal now is to prove, given an alphabet Σ, given $l \in \mathbb{N}_{>0}$ and $u_1, \ldots, u_k \in \Sigma^+$ ($k \in \mathbb{N}_{>0}$) such that for each $i \in [k]$, the letters in u_i are all distinct, that for any $\alpha \in [l]^k$, the language $R^\alpha_l(u_1, \ldots, u_k) \cap S^\alpha_l(u_1, \ldots, u_k)$ is in $\mathcal{P}(\mathbf{J})$; closure of $\mathcal{P}(\mathbf{J})$ under union (Proposition 2.1) consequently entails that $[u_1, \ldots, u_k]_l \in \mathcal{P}(\mathbf{J})$. The way $R^\alpha_l(u_1, \ldots, u_k)$ and $S^\alpha_l(u_1, \ldots, u_k)$ are defined allows us to reason as follows. For each $i \in [k]$ verifying $\alpha_i < l$, let L_i be the language of words w over Σ containing $x_{i,1} u_i{}^{\alpha_i} x_{i,2}$ as a subword but not $x_{i,1} u_i{}^{\alpha_i+1} x_{i,2}$ and such that $w = y_1 u_i y_2$ with $y_1 \in x_{i,1} \sqcup\!\sqcup \Sigma^*$ and $y_2 \in x_{i,2} \sqcup\!\sqcup \Sigma^*$, where $x_{i,1} = u_1{}^{\alpha_1} \cdots u_{i-1}{}^{\alpha_{i-1}}$ and $x_{i,2} = u_{i+1}{}^{\alpha_{i+1}} \cdots u_k{}^{\alpha_k}$. If we manage to prove

that for each $i \in [k]$ verifying $\alpha_i < l$ we have $L_i \in \mathcal{P}(\mathbf{J})$, we can conclude that $R_l^\alpha(u_1, \ldots, u_k) \cap S_l^\alpha(u_1, \ldots, u_k) = (u_1{}^{\alpha_1} \cdots u_k{}^{\alpha_k}) \sqcup \Sigma^* \cap \bigcap_{i \in [k], \alpha_i < l} L_i$ does belong to $\mathcal{P}(\mathbf{J})$ by closure of $\mathcal{P}(\mathbf{J})$ under intersection, Proposition 2.1. The lemma that follows, the main lemma in the proof of Theorem 4.5, exactly shows this. The proof crucially uses the "feedback sweeping" technique, but note that we actually don't know how to prove it when we do not enforce that for each $i \in [k]$, the letters in u_i are all distinct.

Lemma 4.10. *Let Σ be an alphabet and $u \in \Sigma^+$ such that its letters are all distinct. For all $\alpha \in \mathbb{N}_{>0}$ and $x_1, x_2 \in \Sigma^*$, we have*

$$(x_1 u^\alpha x_2) \sqcup \Sigma^* \cap \left((x_1 u^{\alpha+1} x_2) \sqcup \Sigma^*\right)^{\complement} \cap (x_1 \sqcup \Sigma^*) u (x_2 \sqcup \Sigma^*) \in \mathcal{P}(\mathbf{J}) \ .$$

Sketch. Let Σ be an alphabet and $u \in \Sigma^+$ such that its letters are all distinct. Let $\alpha \in \mathbb{N}_{>0}$ and $x_1, x_2 \in \Sigma^*$. We let

$$L = (x_1 u^\alpha x_2) \sqcup \Sigma^* \cap \left((x_1 u^{\alpha+1} x_2) \sqcup \Sigma^*\right)^{\complement} \cap (x_1 \sqcup \Sigma^*) u (x_2 \sqcup \Sigma^*) \ .$$

If $|u| = 1$, the lemma follows trivially because L is piecewise testable and hence belongs to $\mathcal{L}(\mathbf{J})$, so we assume $|u| > 1$.

For each letter $a \in \Sigma$, we shall use $2|u| - 1$ distinct decorated letters of the form $a^{(i)}$ for some $i \in [\![0, 2|u| - 2]\!]$, using the convention that $a^{(0)} = a$; of course, for two distinct letters $a, b \in \Sigma$, we have that $a^{(i)}$ and $b^{(j)}$ are distinct for all $i, j \in [\![0, 2|u| - 2]\!]$. We denote by A the alphabet of these decorated letters. The main idea of the proof is, for a given input length $n \in \mathbb{N}$, to build an A-program Ψ_n over Σ^n such that, given an input word $w \in \Sigma^n$, it first ouputs the $|u| - 1$ first letters of w and then, for each i going from $|u|$ to n, outputs w_i, followed by $w_{i-1}^{(1)} \cdots w_{i-|u|+1}^{(|u|-1)}$ (a "sweep" of $|u|-1$ letters backwards down to position $i-|u|+1$, decorating the letters incrementally) and finally by $w_{i-|u|+2}^{(|u|)} \cdots w_i^{(2|u|-2)}$ (a "sweep" forwards up to position i, continuing the incremental decoration of the letters). The idea behind this way of rearranging and decorating letters is that, given an input word $w \in \Sigma^n$, as long as we make sure that w and thus $\Psi_n(w)$ do contain $x_1 u^\alpha x_2$ as a subword but not $x_1 u^{\alpha+1} x_2$, then $\Psi_n(w)$ can be decomposed as $\Psi_n(w) = y_1 z y_2$ where $y_1 \in x_1 \sqcup \Sigma^*$, $y_2 \in x_2 \sqcup \Sigma^*$, and $|y_1|, |y_2|$ are minimal, with z containing $u^\beta u_{|u|-1}^{(1)} \cdots u_1^{(|u|-1)} u_2^{(|u|)} \cdots u_{|u|}^{(2|u|-2)} u^{\alpha-\beta}$ as a subword for some $\beta \in [\alpha]$ if and only if $w \in (x_1 \sqcup \Sigma^*) u (x_2 \sqcup \Sigma^*)$. This means we can check whether $w \in L$ by testing whether w belongs to some fixed piecewise testable language over A.

The full proof can be found in Appendix B. □

As explained before stating the previous lemma, we can now use it to prove the result we were aiming for. The detailed proof can be found in Appendix B.

Proposition 4.11. *Let Σ be an alphabet, $l \in \mathbb{N}_{>0}$ and $u_1, \ldots, u_k \in \Sigma^+$ ($k \in \mathbb{N}_{>0}$) such that for each $i \in [k]$, the letters in u_i are all distinct. For all $\alpha \in [l]^k$, we have $R_l^\alpha(u_1, \ldots, u_k) \cap S_l^\alpha(u_1, \ldots, u_k) \in \mathcal{P}(\mathbf{J})$.*

We thus derive the awaited corollary.

Corollary 4.12. *Let Σ be an alphabet, $l \in \mathbb{N}_{>0}$ and $u_1, \ldots, u_k \in \Sigma^+$ ($k \in \mathbb{N}_{>0}$) such that for each $i \in [k]$, the letters in u_i are all distinct. Then, $[u_1, \ldots, u_k]_l \in \mathcal{P}(\mathbf{J})$.*

However, what we really want to obtain is that $[u_1, \ldots, u_k]_l \in \mathcal{P}(\mathbf{J})$ without putting any restriction on the u_i's. But, in fact, to remove the constraint that the letters must be all distinct in each of the u_i's, we simply have to decorate each of the input letters with its position minus 1 modulo a big enough $d \in \mathbb{N}_{>0}$. This finally leads to the following proposition; as usual, the proof is in Appendix B.

Proposition 4.13. *Let Σ be an alphabet, $l \in \mathbb{N}_{>0}$ and $u_1, \ldots, u_k \in \Sigma^+$ ($k \in \mathbb{N}_{>0}$). Then $[u_1, \ldots, u_k]_l \in \mathcal{P}(\mathbf{J})$.*

This finishes to prove Theorem 4.5 by closure of $\mathcal{P}(\mathbf{J})$ under Boolean combinations (Proposition 2.1) and by the discussion at the beginning of Subsection 4.1.

5 Conclusion

Although $\mathcal{P}(\mathbf{J})$ is very small compared to AC^0, we have shown that programs over monoids in $\mathbf{J}$ are an interesting subject of study in that they allow to do quite unexpected things. The "feedback-sweeping" technique allows one to detect presence of a factor thanks to such programs as long as this factor does not appear too often as a subword: this is the basic principle behind threshold dot-depth one languages, that our article shows to belong wholly to $\mathcal{P}(\mathbf{J})$.

Whether threshold dot-depth one languages with additional positional modular counting do correspond exactly to the languages in $\mathcal{L}(\mathbf{QDA}) \cap \mathcal{L}(\mathbf{Q}(\mathbf{J} * \mathbf{D}))$ seems to be a challenging question, that we leave open. In his Ph.D. thesis [7], the author proved that all strongly unambiguous monomials (the basic building blocks in $\mathcal{L}(\mathbf{DA})$) that are imposed to belong to $\mathcal{L}(\mathbf{J} * \mathbf{D})$ at the same time are in fact threshold dot-depth one languages. However, the proof looks much too complex and technical to be extended to, say, all languages in $\mathcal{L}(\mathbf{DA}) \cap \mathcal{L}(\mathbf{J} * \mathbf{D})$. New techniques are probably needed, and we might conclude by saying that proving (or disproving) this conjecture could be a nice research goal in algebraic automata theory.

Acknowledgements The author thanks the anonymous referees for their helpful comments and suggestions.

References

[1] M. Ajtai. Σ^1_1-formulae on finite structures. *Annals of pure and applied logic*, 24(1):1–48, 1983.

[2] D. A. M. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *J. Comput. Syst. Sci.*, 38(1):150–164, 1989.

[3] D. A. M. Barrington and D. Thérien. Finite monoids and the fine structure of NC^1. *J. ACM*, 35(4):941–952, 1988.

[4] S. Eilenberg. *Automata, Languages, and Machines*, volume A. Academic Press, New York, 1974.

[5] S. Eilenberg. *Automata, Languages, and Machines*, volume B. Academic Press, New York, 1976.

[6] M. L. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[7] N. Grosshans. *The limits of Nečiporuk's method and the power of programs over monoids taken from small varieties of finite monoids.* PhD thesis, University of Paris-Saclay, France, 2018.

[8] N. Grosshans, P. McKenzie, and L. Segoufin. The power of programs over monoids in DA. In *MFCS 2017, August 21-25, 2017 - Aalborg, Denmark*, pages 2:1–2:20, 2017.

[9] O. Klíma and L. Polák. Hierarchies of piecewise testable languages. *Int. J. Found. Comput. Sci.*, 21(4):517–533, 2010.

[10] C. Lautemann, P. Tesson, and D. Thérien. An algebraic point of view on the Crane Beach property. In *CSL 2006, Szeged, Hungary, September 25-29, 2006*, pages 426–440, 2006.

[11] A. Maciel, P. Péladeau, and D. Thérien. Programs over semigroups of dot-depth one. *Theor. Comput. Sci.*, 245(1):135–148, 2000.

[12] P. McKenzie, P. Péladeau, and D. Thérien. NC^1: The automata-theoretic viewpoint. *Computational Complexity*, 1:330–359, 1991.

[13] P. Péladeau. *Classes de circuits booléens et variétés de monoïdes.* PhD thesis, Université Pierre-et-Marie-Curie (Paris-VI), Paris, France, 1990.

[14] P. Péladeau, H. Straubing, and D. Thérien. Finite semigroup varieties defined by programs. *Theor. Comput. Sci.*, 180(1-2):325–339, 1997.

[15] J. Pin. *Varieties Of Formal Languages.* Plenum Publishing Co., 1986.

[16] J. Pin. The dot-depth hierarchy, 45 years later. In *The Role of Theory in Computer Science - Essays Dedicated to Janusz Brzozowski*, pages 177–202, 2017.

[17] J. Pin and H. Straubing. Some results on $\mathcal{C}$-varieties. *ITA*, 39(1):239–262, 2005.

[18] I. Simon. Piecewise testable events. In *Automata Theory and Formal Languages, 2nd GI Conference, Kaiserslautern, May 20-23, 1975*, pages 214–222, 1975.

[19] H. Straubing. Finite semigroup varieties of the form $V * D$. *Journal of Pure and Applied Algebra*, 36:53–94, 1985.

[20] H. Straubing. When can one finite monoid simulate another? In *Algorithmic Problems in Groups and Semigroups*, pages 267–288. Springer, 2000.

[21] H. Straubing. Languages defined with modular counting quantifiers. *Inf. Comput.*, 166(2):112–132, 2001.

[22] P. Tesson. *Computational Complexity Questions Related to Finite Monoids and Semigroups*. PhD thesis, McGill University, Montreal, 2003.

[23] P. Tesson and D. Thérien. The computing power of programs over finite monoids. *J. Autom. Lang. Comb.*, 7(2):247–258, 2001.

A Missing Proofs from Section 3

A.1 Proof of Proposition 3.1

Upper bound. We start with the upper bound. Given $k \in \mathbb{N}$, we define the alphabet $Y_k = \{e, \#\} \cup \{\bot_l, \top_l \mid l \in [k]\}$; we are going to prove that for all $k \in \mathbb{N}$ there exists a language $Z_k \in \mathcal{PT}_{2k+1}(Y_k^*)$ such that for all $\Delta = (\sigma_n)_{n\in\mathbb{N}}$ sequences of k-selectors, there exists a program-reduction from L_Δ to Z_k of length at most $2 \cdot (k+1)^{-k} \cdot n^{k+1}$. To this end, we use the following proposition and the fact that the language of words of length $n \in \mathbb{N}$ of L_Δ is exactly $K_{n',\sigma_{n'}}$ when there exists $n' \in \mathbb{N}$ verifying $n = (k+1) \cdot n'$ and $\emptyset$ otherwise.

Proposition A.1. *For all $k \in \mathbb{N}$ there is a language $Z_k \in \mathcal{PT}_{2k+1}(Y_k^*)$ such that $\varepsilon \notin Z_k$ and for all $n \in \mathbb{N}$ and all k-selectors σ_n over n, we have $K_{n,\sigma_n} = \Psi^{-1}_{(k+1)\cdot n,\sigma_n}(Z_k^{=|\Psi_{(k+1)\cdot n,\sigma_n}|})$ where $\Psi_{(k+1)\cdot n,\sigma_n}$ is a Y_k-program on $\{0,1\}^{(k+1)\cdot n}$ of length at most $2 \cdot (k+1) \cdot n^{k+1}$.*

Proof. We first define by induction on k a family of languages Z_k over the alphabet Y_k. For $k = 0$, set $Z_0 = Y_0^* \# Y_0^*$. For $k \in \mathbb{N}_{>0}$, the language Z_k is the set of words containing each of $\top_k$ and $\bot_k$ exactly once, the first before the latter, and verifying that the factor between the occurrence of $\top_k$ and the occurrence of $\bot_k$ belongs to Z_{k-1}, i.e. $Z_k = Y_{k-1}^* \top_k Z_{k-1} \bot_k Y_{k-1}^*$. A simple induction on k shows that Z_k for $k \in \mathbb{N}$ is defined by the expression

$$Y_{k-1}^* \top_k Y_{k-2}^* \top_{k-1} \cdots Y_1^* \top_2 Y_0^* \top_1 Y_0^* \# Y_0^* \bot_1 Y_0^* \bot_2 Y_1^* \cdots \bot_{k-1} Y_{k-2}^* \bot_k Y_{k-1}^* \ ,$$

hence it belongs to $\mathcal{PT}_{2k+1}(Y_k^*)$ and in particular does not contain the empty word ε.

Fix $n \in \mathbb{N}$. If $n = 0$, the proposition follows trivially since for any k-selector σ_0 over 0, we have $K_{0,\sigma_0} = \emptyset$ and $\varepsilon \notin Z_k$; otherwise, we define by induction on k a Y_k-program $P_k(d, \sigma)$ on $\{0,1\}^{(d+k+1)\cdot n}$ for every k-selector σ over n and every $d \in \mathbb{N}$.

For any $j \in [n]$ and σ a 0-selector over n, which is just a function in $\mathfrak{P}([n])^{\{\varepsilon\}}$, let $h_{j,\sigma} \colon \{0,1\} \to Y_0$ be the function defined by $h_{j,\sigma}(0) = e$ and $h_{j,\sigma}(1) = \begin{cases} \# & \text{if } j \in \sigma(\varepsilon) \\ e & \text{otherwise} \end{cases}$. For all $k \in \mathbb{N}_{>0}$, we also let f_k and g_k be the functions in ${Y_k}^{\{0,1\}}$ defined by $f_k(0) = g_k(0) = e$, $f_k(1) = \top_k$ and $g_k(1) = \bot_k$. Moreover, for any k-selector σ over n, the symbol $\sigma|j$ for $j \in [n]$ denotes the $(k-1)$-selector over n such that for all $\rho' \in [n]^{k-1}$, we have $i \in \sigma|j(\rho')$ if and only if $i \in \sigma((j, \rho'))$.

For $k \in \mathbb{N}_{>0}$, for $d \in \mathbb{N}$ and σ a k-selector over n, the Y_k-program $P_k(d, \sigma)$ on $\{0,1\}^{(d+k+1)\cdot n}$ is the following sequence of instructions:

$$\begin{aligned} &(d \cdot n + 1, f_k) P_{k-1}(d+1, \sigma|1)(d \cdot n + 1, g_k) \\ &\cdots (d \cdot n + n, f_k) P_{k-1}(d+1, \sigma|n)(d \cdot n + n, g_k) \ . \end{aligned}$$

In words, for each position $i \in [\![d \cdot n + 1, d \cdot n + n]\!]$ with a 1 in the $(d+1)$-th block of n letters in the input, the program runs, between the symbols $\top_k$ and $\bot_k$, the program $P_{k-1}(d+1, \sigma|i)$ obtained by induction for $\sigma|i$ the $(k-1)$-selector over n obtained by restricting σ to all vectors in $[n]^k$ whose first coordinate is i.

For $k = 0$, for $d \in \mathbb{N}$ and σ a 0-selector over n, the Y_0-program $P_0(d, \sigma)$ on $\{0,1\}^{(d+1)\cdot n}$ is the following sequence of instructions:

$$(d \cdot n + 1, h_{1,\sigma})(d \cdot n + 2, h_{2,\sigma}) \cdots (d \cdot n + n, h_{n,\sigma}) \ .$$

In words, for each position $i \in [\![d \cdot n + 1, d \cdot n + n]\!]$ with a 1 in the $(d+1)$-th block of n letters in the input, the program outputs # if and only if $(i - d \cdot n)$ does belong to the set $\sigma(\varepsilon)$.

In short, $P_k(d, \sigma)$ is designed so that for any $w \in \{0,1\}^{(d+k+1)\cdot n}$, the word $P_k(d, \sigma)(w)$ belongs to Z_k if and only if the last $(k+1) \cdot n$ letters of w form a word of $K_{n,\sigma}$.

A simple computation shows that for any $k \in \mathbb{N}$, any $d \in \mathbb{N}$ and σ a k-selector over n, the number of instructions in $P_k(d, \sigma)$ is at most $2 \cdot (k+1) \cdot n^{k+1}$.

A simple induction on k shows that for any $k \in \mathbb{N}$ and $d \in \mathbb{N}$, when running on a word $w \in \{0,1\}^{(d+k+1)\cdot n}$, for any σ a k-selector over n, the program $P_k(d, \sigma)$ returns a word in Z_k if and only if when $u^{(1)}, u^{(2)}, \ldots, u^{(k)}, v$ are the last $k+1$ consecutive blocks of n letters of w, then $u^{(1)}, u^{(2)}, \ldots, u^{(k)}$ each contain 1 exactly once and define the vector ρ in $[n]^k$ where for all $i \in [k]$, the value ρ_i is given by the position of the only 1 in $u^{(i)}$, verifying that there exists $j \in \sigma_n(\rho)$ such that v_j is 1.

Therefore, for any $k \in \mathbb{N}$ and σ_n a k-selector over n, if we set $\Psi_{(k+1)\cdot n, \sigma_n} = P_k(0, \sigma_n)$, we have $K_{n,\sigma_n} = \Psi^{-1}_{(k+1)\cdot n, \sigma_n}(Z_k^{=|\Psi_{(k+1)\cdot n, \sigma_n}|})$ where $\Psi_{(k+1)\cdot n, \sigma_n}$ is a Y_k-program on $\{0,1\}^{(k+1)\cdot n}$ of length at most $2 \cdot (k+1) \cdot n^{k+1}$. □

Consequently, for all $k \in \mathbb{N}$ and any sequence of k-selectors Δ, since the language Z_k is in $\mathcal{PT}_{2k+1}(Y_k^*)$ and thus recognised by a monoid from $\mathbf{J_{2k+1}}$, we have, by Proposition 2.2, that $L_\Delta \in \mathcal{P}\big(\mathbf{J_{2k+1}}, n^{k+1}\big)$.

Lower bound. For the lower bound, we use the following claim, whose proof can be found in [8, Claim 10].

Claim A.2. *For all $i \in \mathbb{N}_{>0}$ and $n \in \mathbb{N}$, the number of languages in $\{0,1\}^n$ recognised by programs over a monoid of order i on $\{0,1\}^n$, with at most $l \in \mathbb{N}$ instructions, is upper-bounded by $i^{i^2} 2^i \cdot (n \cdot i^2)^l$.*

If for some $k \in \mathbb{N}$ and $i \in [\alpha]$ with $\alpha \in \mathbb{N}_{>0}$, we apply this claim for all $n \in \mathbb{N}$ and $l = \alpha \cdot ((k+1) \cdot n)^k$, we get a number $\mu_i(n)$ of languages in $\{0,1\}^{(k+1)\cdot n}$ recognised by programs over a monoid of order i on $\{0,1\}^{(k+1)\cdot n}$ with at most l instructions that is in $2^{\mathrm{O}(n^k \log_2(n))}$, which is asymptotically strictly smaller than the number of distinct K_{n,σ_n} when the k-selector σ_n over n varies, which is $2^{n^{k+1}}$, i.e. $\mu_i(n)$ is in $\mathrm{o}(2^{n^{k+1}})$.

Hence, for all $j \in \mathbb{N}_{>0}$, there exist an $n_j \in \mathbb{N}$ and τ_j a k-selector over n_j such that no program over a monoid of order $i \in [j]$ on $\{0,1\}^{(k+1)\cdot n_j}$ and of length at most $j \cdot ((k+1) \cdot n_j)^k$ recognises K_{n_j,τ_j}. Moreover, we can assume without loss of generality that the sequence $(n_j)_{j \in \mathbb{N}_{>0}}$ is increasing. Let $\Delta = (\sigma_n)_{n \in \mathbb{N}}$ be such that $\sigma_{n_j} = \tau_j$ for all $j \in \mathbb{N}_{>0}$ and $\sigma_n \colon [n]^k \to \mathfrak{P}([n]), \rho \mapsto \emptyset$ for any $n \in \mathbb{N}$ verifying that it is not equal to any n_j for $j \in \mathbb{N}_{>0}$. We show that no sequence of programs over a finite monoid of length $\mathrm{O}(n^k)$ can recognise L_Δ. If this were the case, then let i be the order of the monoid. Let $j \in \mathbb{N}, j \geq i$ be such that for any $n \in \mathbb{N}$, the n-th program has length at most $j \cdot n^k$. But, by construction, we

know that there does not exist any such program on $\{0,1\}^{(k+1)\cdot n_j}$ recognising K_{n_j,τ_j}, a contradiction.

This implies that for all $k \in \mathbb{N}$, we have $\mathcal{P}(\mathbf{J}, n^k) \subset \mathcal{P}(\mathbf{J}, n^{k+1})$ and additionally that for all $d \in \mathbb{N}, d \leq \lceil \frac{k}{2} \rceil - 1$, we have $\mathcal{P}(\mathbf{J_k}, n^d) \subset \mathcal{P}(\mathbf{J_k}, n^{d+1})$, since any monoid from $\mathbf{J_d}$ is also a monoid from $\mathbf{J_k}$.

A.2 Proof of Proposition 3.2

Actually, the equivalent shorter program we give is even a *subprogram* of the original one, i.e. a subsequence of the latter. For P some program over a finite monoid M, we may denote by ξ_P the function that associates to each possible input word w the word in $M^{|P|}$ obtained by successively evaluating the instructions of P for w.

Observe that given P a program over some finite monoid M on Σ^n for $n \in \mathbb{N}$ and Σ an alphabet, a subprogram P' of P is equivalent to P if and only if for every language $K \subseteq M^*$ recognised by the evaluation morphism η_M of M, the unique morphism from M^* to M extending the identity on M, we have $\xi_P(w) \in K \Leftrightarrow \xi_{P'}(w) \in K$ for all $w \in \Sigma^n$. Moreover, every language recognised by η_M is precisely a language of $\mathcal{PT}_k(M^*)$ when $M \in \mathbf{J_k}$ for some $k \in \mathbb{N}$.

The result is hence a consequence of the following lemma and the fact that every language in $\mathcal{PT}_k(M^*)$ is a union of $\sim_k$-classes, each of those classes corresponding to all words over M having the same set of k-subwords, that is finite.

Lemma A.3. *Let Σ be an alphabet and M a finite monoid.*

For all $k \in \mathbb{N}$, there exists a constant $c \in \mathbb{N}_{>0}$ verifying that for any program P over M on Σ^n for $n \in \mathbb{N}$ and any word $t \in M^k$, there exists a subprogram Q of P of length at most $c \cdot n^{\lceil k/2 \rceil}$ such that for any subprogram Q' of P that has Q as a subprogram, we have that t is a subword of $\xi_P(w)$ if and only if t is a subword of $\xi_{Q'}(w)$ for all $w \in \Sigma^n$.

Proof. A program P over M on Σ^n for $n \in \mathbb{N}$ is a finite sequence (p_i, f_i) of instructions where each p_i is a positive natural number which is at most n and each f_i is a function from Σ to M. We denote by l the number of instructions of P. For each set $I \subseteq [l]$ we denote by $P[I]$ the subprogram of P consisting of the subsequence of instructions of P obtained after removing all instructions whose index is not in I. When $I = [\![i, j]\!]$ for some $i, j \in [l]$, we may write $P[i, j]$ instead of $P[I]$.

We prove the lemma by induction on k, fixing the constant to be $c_k = k! \cdot |\Sigma|^{\lceil k/2 \rceil}$ for a given $k \in \mathbb{N}$.

The intuition behind the proof for a program P on inputs of length n and some t of length at least 3 is as follows. Given l the length of P, we will select a subset I of the indices of instructions numbered from 1 to l to obtain $P[I]$ verifying the conditions of the lemma. Consider all the indices $1 \leq i_1 < i_2 < \cdots < i_s \leq l$ that each correspond, for some letter a and some position p in the input, to the first instruction of P that would output the element t_1 when reading a at position p or to the last instruction of P that would output the element t_k when reading a at position p. We then have that, given some w as input, t is a subword of $\xi_P(w)$ if and only if there exist $1 \leq \gamma < \delta \leq s$ verifying that the element at position i_γ of $\xi_P(w)$ is t_1, the element at position i_δ of $\xi_P(w)$

is t_k and $t_2 \cdots t_{k-1}$ is a subword of $\xi_{P[i_\gamma+1,i_\delta-1]}(w)$. The idea is then that if we set I to contain $i_1, i_2, \ldots, i_s$ as well as all indices obtained by induction for $P[i_j+1, i_{j+1}-1]$ and $t_\alpha \cdots t_\beta$ for all $1 \leq j \leq s-1$ and $1 < \alpha \leq \beta < k$, we would have that for all w, the word t is a subword of $\xi_P(w)$ if and only if it is a subword of $\xi_{P[I]}(w)$, that is $\xi_P(w)$ where only the elements at indices in I have been kept. The length upper bound of the order of $n^{\lceil k/2 \rceil}$ would be met because the number of possible values for j is $s-1$, hence at most linear in n, and the number of possible values for (α, β) is quadratic in k, a constant.

The intuition behind the proof when t is of length less than 3 is essentially the same, but without induction.

Inductive step. Let $k \in \mathbb{N}, k \geq 3$ and assume the lemma proven for all $k' \in \mathbb{N}, k' < k$. Let P be a program over M on Σ^n for $n \in \mathbb{N}$ of length $l \in \mathbb{N}$ and some word $t \in M^k$.

Observe that when $n = 0$, we necessarily have $P = \varepsilon$, so that the lemma is trivially proven in that case. So we now assume $n > 0$.

For each $p \in [n]$ and each $a \in \Sigma$ consider within the sequence of instructions of P the first instruction of the form (p, f) with $f(a) = t_1$ and the last instruction of that form with $f(a) = t_k$, if they exist. We let $I_{(1,k)}$ be the set of indices of these instructions for all a and p. Notice that the size of $I_{(1,k)}$ is at most $2 \cdot |\Sigma| \cdot n$.

Let $s = \left|I_{(1,k)}\right|$ and let us denote $I_{(1,k)} = \{i_1, i_2, \ldots, i_s\}$ where $i_1 < i_2 < \cdots < i_s$. Given $\alpha, \beta \in [k]$, we also set $t^{(\alpha,\beta)} = t_\alpha t_{\alpha+1} \cdots t_\beta$. For all $\alpha, \beta \in [k]$ such that $1 < \alpha \leq \beta < k$ and $j \in [s-1]$, we let $J_{j,(\alpha,\beta)}$ be the set of indices of the instructions within $P[i_j+1, i_{j+1}-1]$ appearing in its subprogram obtained by induction for $P[i_j+1, i_{j+1}-1]$ and $t^{(\alpha,\beta)}$.

We now let I be the union of $I_{(1,k)}$ and $J'_{j,(\alpha,\beta)} = \{e + i_j \mid e \in J_{j,(\alpha,\beta)}\}$ for all $\alpha, \beta \in [k]$ such that $1 < \alpha \leq \beta < k$ and $j \in [s-1]$ (the translation being required because the first instruction in $P[i_j+1, i_{j+1}-1]$ is the (i_j+1)-th instruction in P). We claim that $Q = P[I]$, a subprogram of P, has the desired properties.

First notice that by induction the size of $J'_{j,(\alpha,\beta)}$ for all $\alpha, \beta \in [k]$ such that $1 < \alpha \leq \beta < k$ and $j \in [s-1]$ is upper bounded by

$$(\beta - \alpha + 1)! \cdot |\Sigma|^{\lceil (\beta-\alpha+1)/2 \rceil} \cdot n^{\lceil (\beta-\alpha+1)/2 \rceil} \leq (k-2)! \cdot |\Sigma|^{\lceil (k-2)/2 \rceil} \cdot n^{\lceil (k-2)/2 \rceil}.$$

Hence, the size of I is at most

$$\begin{aligned}
&\left|I_{(1,k)}\right| + \sum_{j=1}^{s-1} \sum_{1<\alpha\leq\beta<k} \left|J'_{j,(\alpha,\beta)}\right| \\
\leq &2 \cdot |\Sigma| \cdot n + (2 \cdot |\Sigma| \cdot n - 1) \cdot \frac{(k-1)\cdot(k-2)}{2} \cdot (k-2)! \cdot |\Sigma|^{\lceil (k-2)/2 \rceil} \cdot n^{\lceil (k-2)/2 \rceil} \\
\leq &2 \cdot |\Sigma| \cdot n + (2 \cdot |\Sigma| \cdot n - 1) \cdot \frac{k\cdot(k-1)}{2} \cdot (k-2)! \cdot |\Sigma|^{\lceil (k-2)/2 \rceil} \cdot n^{\lceil (k-2)/2 \rceil} \\
\leq &k! \cdot |\Sigma|^{\lceil k/2 \rceil} \cdot n^{\lceil k/2 \rceil} = c_k \cdot n^{\lceil k/2 \rceil}
\end{aligned}$$

as $\left|\{(\alpha, \beta) \in \mathbb{N}^2 \mid 1 < \alpha \leq \beta < k\}\right| = \sum_{j=2}^{k-1}(k-j) = \sum_{j=1}^{k-2} j = \frac{(k-1)\cdot(k-2)}{2}$ and $2 \cdot |\Sigma| \cdot n \leq \frac{k!}{2} \cdot |\Sigma|^{\lceil (k-2)/2 \rceil} \cdot n^{\lceil (k-2)/2 \rceil}$ since $k \geq 3$, so that $P[I]$ has at most the required length.

Let Q' be a subprogram of P that has Q as a subprogram: it means that there exists some set $I' \subseteq [l]$ containing I such that $Q' = P[I']$.

Take $w \in \Sigma^n$.

Assume now that t is a subword of $\xi_P(w)$. It means that there exist $r_1, r_2, \ldots, r_k \in [l]$, $r_1 < r_2 < \cdots < r_k$, such that for all $j \in [k]$, we have $f_{r_j}(w_{p_{r_j}}) = t_j$. By definition of $I_{(1,k)}$, there exist $\gamma, \delta \in [s], \gamma < \delta$, such that $i_\gamma \leq r_1 < r_k \leq i_\delta$ and $f_{i_\gamma}(w_{p_{i_\gamma}}) = t_1$ and $f_{i_\delta}(w_{p_{i_\delta}}) = t_k$. For each $j \in [\![\gamma, \delta - 1]\!]$, let $m_j \in [\![2, k]\!]$ be the smallest integer in $[\![2, k-1]\!]$ such that $i_j \leq r_{m_j} < i_{j+1}$ and k if it does not exist, and $M_j \in [\![1, k-1]\!]$ be the biggest integer in $[\![2, k-1]\!]$ such that $i_j \leq r_{M_j} < i_{j+1}$ and 1 if it does not exist. Observe that, since for each $j \in [\![\gamma, \delta - 1]\!]$, we have $t^{(m_j, M_j)} = t^{(k,1)} = \varepsilon$ if there does not exist any $o \in [\![2, k-1]\!]$ verifying $i_j \leq r_o < i_{j+1}$, it holds that $t^{(2,k-1)} = \prod_{j=\gamma}^{\delta-1} t^{(m_j, M_j)}$. For all $j \in [\![\gamma, \delta - 1]\!]$, we have that for any set $J \subseteq [i_{j+1} - i_j - 1]$ containing $\bigcup_{1<\alpha\leq\beta<k} J_{j,(\alpha,\beta)}$, the word $t^{(m_j, M_j)}$ is a subword of $f_{i_j}(w_{p_{i_j}})\xi_{P[i_j+1, i_{j+1}-1][J]}(w)$ when $m_j < k$ and $r_{m_j} = i_j$, and of $\xi_{P[i_j+1, i_{j+1}-1][J]}(w)$ otherwise. Indeed, let $j \in [\![\gamma, \delta - 1]\!]$.

- If $m_j < k$ and $r_{m_j} = i_j$, then $f_{i_j}(w_{p_{i_j}}) = f_{r_{m_j}}(w_{p_{r_{m_j}}}) = t_{m_j}$ and $i_j = r_{m_j} < r_{m_j+1} < \cdots < r_{M_j} < i_{j+1}$, so $t^{(m_j+1, M_j)}$ is a subword of $\xi_{P[i_j+1, i_{j+1}-1]}(w)$. This implies, directly when $m_j = M_j$ or by induction otherwise, that for any set $J \subseteq [i_{j+1} - i_j - 1]$ containing $\bigcup_{1<\alpha\leq\beta<k} J_{j,(\alpha,\beta)}$, the word $t^{(m_j+1, M_j)}$ is a subword of $\xi_{P[i_j+1, i_{j+1}-1][J]}(w)$. This implies in turn that $t^{(m_j, M_j)}$ is a subword of $f_{i_j}(w_{p_{i_j}})\xi_{P[i_j+1, i_{j+1}-1][J]}(w)$.

- Otherwise, when $m_j = k$, there does not exist any $o \in [\![2, k-1]\!]$ verifying $i_j \leq r_o < i_{j+1}$, so $t^{(m_j, M_j)} = \varepsilon$ is trivially a subword of $\xi_{P[i_j+1, i_{j+1}-1][J]}(w)$ for any set $J \subseteq [i_{j+1} - i_j - 1]$ containing $\bigcup_{1<\alpha\leq\beta<k} J_{j,(\alpha,\beta)}$. And when $m_j < k$ but $r_{m_j} \neq i_j$, it means that $r_{m_j} > i_j$, hence $i_j < r_{m_j} < r_{m_j+1} < \cdots < r_{M_j} < i_{j+1}$, so $t^{(m_j, M_j)}$ is a subword of $\xi_{P[i_j+1, i_{j+1}-1]}(w)$. This implies, by induction, that $t^{(m_j, M_j)}$ is a subword of $\xi_{P[i_j+1, i_{j+1}-1][J]}(w)$ for any set $J \subseteq [i_{j+1} - i_j - 1]$ containing $\bigcup_{1<\alpha\leq\beta<k} J_{j,(\alpha,\beta)}$.

Therefore, using the convention that $i_0 = 0$ and $i_{s+1} = l + 1$, if we define, for each $j \in [\![0, s]\!]$, the set $I'_j = \{e - i_j \mid e \in I', i_j < e < i_{j+1}\}$ as the subset of I' of elements strictly between i_j and i_{j+1} translated by $-i_j$, we have that $t^{(2,k-1)}$ is a subword of

$$\xi_{P[i_\gamma+1, i_{\gamma+1}-1][I'_\gamma]}(w) f_{i_{\gamma+1}}(w_{p_{i_{\gamma+1}}}) \xi_{P[i_{\gamma+1}+1, i_{\gamma+2}-1][I'_{\gamma+1}]}(w) \cdots f_{i_{\delta-1}}(w_{p_{i_{\delta-1}}}) \xi_{P[i_{\delta-1}+1, i_\delta-1][I'_{\delta-1}]}(w)$$

(since we have $r_{m_\gamma} \geq r_2 > r_1 \geq i_\gamma$), so that, as $f_{i_\gamma}(w_{p_{i_\gamma}}) = t_1$ and $f_{i_\delta}(w_{p_{i_\delta}}) = t_k$, we have that $t = t_1 t^{(2,k-1)} t_k$ is a subword of

$$\begin{aligned} &\xi_{P[1, i_1-1][I'_0]}(w) f_{i_1}(w_{p_{i_1}}) \xi_{P[i_1+1, i_2-1][I'_1]}(w) \cdots f_{i_s}(w_{p_{i_s}}) \xi_{P[i_s+1, l][I'_s]}(w) \\ =&\xi_{P[I']}(w) \ . \end{aligned}$$

Assume finally that t is a subword of $\xi_{P[I']}(w)$. Then it is obviously a subword of $\xi_P(w)$, as $\xi_{P[I']}(w)$ is a subword of $\xi_P(w)$.

Therefore, t is a subword of $\xi_P(w)$ if and only if t is a subword of $\xi_{Q'}(w) = \xi_{P[I']}(w)$, as desired.

Base case. There are three subcases to consider.

Subcase $k = 2$. Let P be a program over M on Σ^n for $n \in \mathbb{N}$ of length $l \in \mathbb{N}$ and some word $t \in M^2$.

We use the same idea as in the inductive step.

Observe that when $n = 0$, we necessarily have $P = \varepsilon$, so that the lemma is trivially proven in that case. So we now assume $n > 0$.

For each $p \in [n]$ and each $a \in \Sigma$ consider within the sequence of instructions of P the first instruction of the form (p, f) with $f(a) = t_1$ and the last instruction of that form with $f(a) = t_2$, if they exist. We let I be the set of indices of these instructions for all a and p. Notice that the size of I is at most $2 \cdot |\Sigma| \cdot n = 2! \cdot |\Sigma|^{\lceil 2/2 \rceil} \cdot n^{\lceil 2/2 \rceil} = c_2 \cdot n^{\lceil 2/2 \rceil}$.

We claim that $Q = P[I]$, a subprogram of P, has the desired properties. We just showed it has at most the required length.

Let Q' be a subprogram of P that has Q as a subprogram: it means that there exists some set $I' \subseteq [l]$ containing I such that $Q' = P[I']$.

Take $w \in \Sigma^n$.

Assume now that t is a subword of $\xi_P(w)$. It means there exist $i_1, i_2 \in [l], i_1 < i_2$ such that $f_{i_1}(w_{p_{i_1}}) = t_1$ and $f_{i_2}(w_{p_{i_2}}) = t_2$. By definition of I, there exist ${i_1}', {i_2}' \in I$, such that ${i_1}' \leq i_1 < i_2 \leq {i_2}'$ and $f_{{i_1}'}(w_{p_{{i_1}'}}) = t_1$ and $f_{{i_2}'}(w_{p_{{i_2}'}}) = t_2$. Hence, as $f_{{i_1}'}(w_{p_{{i_1}'}}) f_{{i_2}'}(w_{p_{{i_2}'}})$ is a subword of $\xi_{P[I']}(w)$ (because $I \subseteq I'$), we get that $t = t_1 t_2$ is a subword of $\xi_{P[I']}(w)$.

Assume finally that t is a subword of $\xi_{P[I']}(w)$. Then it is obviously a subword of $\xi_P(w)$, as $\xi_{P[I']}(w)$ is a subword of $\xi_P(w)$.

Therefore, t is a subword of $\xi_P(w)$ if and only if t is a subword of $\xi_{Q'}(w) = \xi_{P[I']}(w)$, as desired.

Subcase $k = 1$. Let P be a program over M on Σ^n for $n \in \mathbb{N}$ of length $l \in \mathbb{N}$ and some word $t \in M^1$.

We again use the same idea as before.

Observe that when $n = 0$, we necessarily have $P = \varepsilon$, so that the lemma is trivially proven in that case. So we now assume $n > 0$.

For each $p \in [n]$ and each $a \in \Sigma$ consider within the sequence of instructions of P the first instruction of the form (p, f) with $f(a) = t_1$, if it exists. We let I be the set of indices of these instructions for all a and p. Notice that the size of I is at most $|\Sigma| \cdot n = 1! \cdot |\Sigma|^{\lceil 1/2 \rceil} \cdot n^{\lceil 1/2 \rceil} = c_1 \cdot n^{\lceil 1/2 \rceil}$.

We claim that $Q = P[I]$, a subprogram of P, has the desired properties. We just showed it has at most the required length.

Let Q' be a subprogram of P that has Q as a subprogram: it means that there exists some set $I' \subseteq [l]$ containing I such that $Q' = P[I']$.

Take $w \in \Sigma^n$.

Assume now that t is a subword of $\xi_P(w)$. It means there exists $i \in [l]$ such that $f_i(w_{p_i}) = t_1$. By definition of I, there exists $i' \in I$ such that $i' \leq i$ and $f_{i'}(w_{p_{i'}}) = t_1$. Hence, as $f_{i'}(w_{p_{i'}})$ is a subword of $\xi_{P[I']}(w)$ (because $I' \subseteq I$), we get that $t = t_1$ is a subword of $\xi_{P[I']}(w)$.

Assume finally that t is a subword of $\xi_{P[I']}(w)$. Then it is obviously a subword of $\xi_P(w)$, as $\xi_{P[I']}(w)$ is a subword of $\xi_P(w)$.

Therefore, t is a subword of $\xi_P(w)$ if and only if t is a subword of $\xi_{Q'}(w) = \xi_{P[I']}(w)$, as desired.

Subcase $k = 0$. Let P be a program over M on Σ^n for $n \in \mathbb{N}$ of length $l \in \mathbb{N}$ and some word $t \in M^0$.

We claim that $Q = \varepsilon$, a subprogram of P, has the desired properties.

First notice that the length of Q is $0 \leq 0! \cdot |\Sigma|^{\lceil 0/2 \rceil} \cdot n^{\lceil 0/2 \rceil} = c_0 \cdot n^{\lceil 0/2 \rceil}$, at most the required length.

Let Q' be a subprogram of P that has Q as a subprogram. As $t \in M^0$, we necessarily have that $t = \varepsilon$, which is a subword of any word in M^*. Therefore, we immediately get that for all $w \in \Sigma^n$, the word t is a subword of $\xi_P(w)$ if and only if t is a subword of $\xi_{Q'}(w)$, as desired. □

B Missing Proofs from Section 4

B.1 Proof of Lemma 4.1

Proof of Lemma 4.1. Let $\Sigma = \{a, b, c\}$.

Let

$$L = ca \sqcup\!\sqcup \Sigma^* \cap (cca \sqcup\!\sqcup \Sigma^*)^{\complement} \cap (caa \sqcup\!\sqcup \Sigma^*)^{\complement} \cap (cb \sqcup\!\sqcup \Sigma^*)^{\complement}$$

be the language of all words over Σ having ca as a subword but not the subwords cca, caa and cb, that by construction is piecewise testable, i.e. belongs to $\mathcal{L}(\mathbf{J})$.

We are now going to build a program-reduction from $(a+b)^*ac^+$ to L. Let $n \in \mathbb{N}$. If $n \leq 1$, we set Ψ_n to be ε, the empty Σ-program on Σ^n. Otherwise, if $n \geq 2$, we set

$$\Psi_n = (2, \mathrm{id}_\Sigma)(1, \mathrm{id}_\Sigma)(3, \mathrm{id}_\Sigma)(2, \mathrm{id}_\Sigma)(4, \mathrm{id}_\Sigma)(3, \mathrm{id}_\Sigma) \cdots (n, \mathrm{id}_\Sigma)(n-1, \mathrm{id}_\Sigma) \ .$$

Let us define $s \colon \mathbb{N} \to \mathbb{N}$ by $s(n) = |\Psi_n|$ for all $n \in \mathbb{N}$, which is such that

$$s(n) = \begin{cases} 0 & \text{if } n \leq 1 \\ 2n-2 & \text{otherwise } (n \geq 2) \end{cases}$$

for all $n \in \mathbb{N}$. Fix $n \in \mathbb{N}$.

Let $w \in ((a+b)^*ac^+)^{=n}$: it means $n \geq 2$ and there exist $u \in (a+b)^{n_1}$ with $n_1 \in [\![0, n-2]\!]$ and $n_2 \in [\![0, n-2]\!]$ verifying that $w = uacc^{n_2}$ and $n_1 + n_2 = n-2$. We therefore have

$$\Psi_n(w) = \begin{cases} cac^{2n_2} & \text{when } n_1 = 0 \\ u_2 u_1 \cdots u_{n_1} u_{n_1 - 1} a u_{n_1} c a c^{2n_2} & \text{otherwise } (n_1 > 0) \ , \end{cases}$$

a word easily seen to belong to $L^{=2n-2}$. Since this is true for all $w \in ((a+b)^*ac^+)^{=n}$, it follows that $((a+b)^*ac^+)^{=n} \subseteq \Psi_n^{-1}(L^{=s(n)})$.

Let conversely $w \in \Psi_n^{-1}(L^{=s(n)})$. Since this means that $\Psi_n(w) \in L^{=s(n)}$, we necessarily have $n \geq 2$ as it must contain ca as a subword, so that

$$\Psi_n(w) = w_2 w_1 w_3 w_2 w_4 w_3 \cdots w_n w_{n-1} \ .$$

Let $i, j \in [n]$ verifying that $w_i = c$, that $w_j = a$ and $w_i w_j$ is a subword of $\Psi_n(w)$. This means that $j \geq i-1$, and we will now show that, actually, $j = i-1$. Assume that $j \geq i+2$; by construction, this would mean that $w_i w_j w_j = caa$ is a subword of $\Psi_n(w)$, a contradiction to the fact it belongs to L. Assume otherwise that $j = i+1$; by construction, this would either mean that $w_i w_{i-1} w_{i+1} w_i$ is a subword of $\Psi_n(w)$, which would imply one of caa, cba and cca is a subword of $\Psi_n(w)$, or that $w_{i+1} w_i w_{i+2} w_{i+1}$ is a subword of $\Psi_n(w)$, which would imply one

of caa, cba and cca is a subword of $\Psi_n(w)$, in both cases contradicting the fact $\Psi_n(w)$ belongs to L. Hence, we indeed have $j = i-1$, and in particular that $i \geq 2$. Now, by construction, for each $t \in [i-2]$, we have that $w_t w_i w_{i-1} = w_t ca$ is a subword of $\Psi_n(w) \in L$, so that w_t cannot be equal to c. Similarly, for each $t \in [\![i+1, n]\!]$, we have that $w_i w_{i-1} w_t = caw_t$ is a subword of $\Psi_n(w) \in L$, so that w_t must be equal to c. This means that $w_1 \cdots w_{i-2} \in (a+b)^*$ and $w_{i+1} \cdots w_n \in c^*$, so that $w \in (a+b)^* acc^* = (a+b)^* ac^+$. Since this is true for all $w \in \Psi_n^{-1}(L^{=s(n)})$, it follows that $((a+b)^* ac^+)^{=n} \supseteq \Psi_n^{-1}(L^{=s(n)})$.

Therefore, we have that $((a+b)^* ac^+)^{=n} = \Psi_n^{-1}(L^{=s(n)})$ for all $n \in \mathbb{N}$, so $(\Psi_n)_{n\in\mathbb{N}}$ is a program reduction from $(a+b)^* ac^+$ to L of length $s(n)$. So since $L \in \mathcal{L}(\mathbf{J})$, we can conclude that $(a+b)^* ac^+ \in \mathcal{P}(\mathbf{J}, s(n)) = \mathcal{P}(\mathbf{J}, n)$ by Proposition 2.2. □

B.2 Proof of Lemma 4.9

Proof of Lemma 4.9. Let Σ be an alphabet and $l \in \mathbb{N}_{>0}$. We prove it by induction on $k \in \mathbb{N}_{>0}$.

Base case $k = 1$. Let $u_1 \in \Sigma^+$ such that the letters in u_1 are all distinct. It is clear that

$$\begin{aligned}
&\bigcup_{q_1 \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \\
=&(\Sigma^* u_1 \Sigma^* \cup {u_1}^l \sqcup\!\sqcup \Sigma^*) \\
=&\Big(\bigcup_{\alpha_1=1}^{l-1} \big({u_1}^{\alpha_1} \sqcup\!\sqcup \Sigma^* \cap ({u_1}^{\alpha_1+1} \sqcup\!\sqcup \Sigma^*)^{\complement} \cap \Sigma^* u_1 \Sigma^* \big) \cup ({u_1}^l \sqcup\!\sqcup \Sigma^*) \Big) \\
=& \bigcup_{\alpha_1 \in [l]} \big(R_l^{\alpha_1}(u_1) \cap S_l^{\alpha_1}(u_1) \big) \ .
\end{aligned}$$

Induction. Let $k \in \mathbb{N}_{>0}$ and assume that for all $u_1, \ldots, u_k \in \Sigma^+$ such that for each $i \in [k]$, the letters in u_i are all distinct, we have

$$\bigcup_{q_1,\ldots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)} = \bigcup_{\alpha \in [l]^k} \big(R_l^{\alpha}(u_1, \ldots, u_k) \cap S_l^{\alpha}(u_1, \ldots, u_k) \big) \ .$$

Let now $u_1, \ldots, u_{k+1} \in \Sigma^+$ such that for each $i \in [k+1]$, the letters in u_i are all distinct.

Right-to-left inclusion. Let

$$w \in \bigcup_{\alpha \in [l]^{k+1}} \big(R_l^{\alpha}(u_1, \ldots, u_{k+1}) \cap S_l^{\alpha}(u_1, \ldots, u_{k+1}) \big) \ .$$

Let $\alpha \in [l]^{k+1}$ witnessing this fact. As $w \in R_l^{\alpha}(u_1, \ldots, u_{k+1})$, we can decompose it as $w = xy$ where $x \in ({u_1}^{\alpha_1} \cdots {u_k}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*$ and $y \in {u_{k+1}}^{\alpha_{k+1}} \sqcup\!\sqcup \Sigma^*$ with $|y|$ being minimal. What we are going to do is, on the one hand, to prove that $x \in R_l^{\alpha'}(u_1, \ldots, u_k) \cap S_l^{\alpha'}(u_1, \ldots, u_k)$ where $\alpha' = (\alpha_1, \ldots, \alpha_k)$, so that we can apply the inductive hypothesis on x and get that there exist $q_1, \ldots, q_k \in \{1, l\}$ such that $x \in L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)}$; and, on the other hand, we are going to prove

that there exists $q_{k+1} \in \{1, l\}$ verifying $y \in L^{(l)}_{(u_{k+1}, q_{k+1})}$. We now spell out the details.

For each $i \in [k], \alpha_i < l$, we have $x \notin ({u_1}^{\alpha_1} \cdots {u_i}^{\alpha_i+1} \cdots {u_k}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*$, otherwise we would have $w = xy \in ({u_1}^{\alpha_1} \cdots {u_i}^{\alpha_i+1} \cdots {u_{k+1}}^{\alpha_{k+1}}) \sqcup\!\sqcup \Sigma^*$. Also, for all $i \in [k], \alpha_i < l$, we have that $x \in \big(({u_1}^{\alpha_1} \cdots {u_{i-1}}^{\alpha_{i-1}}) \sqcup\!\sqcup \Sigma^*\big) u_i \big(({u_{i+1}}^{\alpha_{i+1}} \cdots {u_k}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big)$, otherwise it would mean that $y = y_1 y_2$ with $|y_1| > 0$, that $xy_1 \in \big(({u_1}^{\alpha_1} \cdots {u_{i-1}}^{\alpha_{i-1}}) \sqcup\!\sqcup \Sigma^*\big) u_i \big(({u_{i+1}}^{\alpha_{i+1}} \cdots {u_k}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big)$ and $y_2 \in {u_{k+1}}^{\alpha_{k+1}} \sqcup\!\sqcup \Sigma^*$, contradicting the minimality of $|y|$. So $x \in R_l^{\alpha'}(u_1, \ldots, u_k) \cap S_l^{\alpha'}(u_1, \ldots, u_k)$, which means by inductive hypothesis that there exist $q_1, \ldots, q_k \in \{1, l\}$ such that $x \in L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)}$.

Remember now that the letters in u_{k+1} are all distinct. If $\alpha_{k+1} < l$, since $w \in \big(({u_1}^{\alpha_1} \ldots {u_k}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big) u_{k+1} \Sigma^*$, we must have $y \in \Sigma^* u_{k+1} \Sigma^*$. Indeed, by minimality of $|y|$, the word y starts with the first letter of u_{k+1}, which has pairwise distinct letters, so that u_{k+1} cannot appear as a factor of xy partly in x and partly in y; so if it were the case that y does not contain u_{k+1} as a factor, we would have $x \in \big(({u_1}^{\alpha_1} \ldots {u_k}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big) u_{k+1} \Sigma^*$, so that $xy = w \in ({u_1}^{\alpha_1} \ldots {u_k}^{\alpha_k} {u_{k+1}}^{\alpha_{k+1}+1}) \sqcup\!\sqcup \Sigma^*$, a contradiction with the hypothesis on w. Hence, $y \in L^{(l)}_{(u_{k+1},\alpha_{k+1})}$. If $\alpha_{k+1} = l$, then $y \in {u_{k+1}}^{\alpha_{k+1}} \sqcup\!\sqcup \Sigma^* = L^{(l)}_{(u_{k+1},\alpha_{k+1})}$. So, if we set $q_{k+1} = \begin{cases} 1 & \text{if } \alpha_{k+1} < l \\ l & \text{otherwise} \end{cases}$, then we get that $y \in L^{(l)}_{(u_{k+1},q_{k+1})}$.

We can conclude that $w = xy \in L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)} L^{(l)}_{(u_{k+1},q_{k+1})}$.

Left-to-right inclusion. Let $w \in \bigcup_{q_1,\ldots,q_{k+1} \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_{k+1},q_{k+1})}$. The rough idea of our proof here is to take $\alpha_{k+1} \in [l]$ the biggest integer in $[l]$ such that $w \in \big(\bigcup_{q_1,\ldots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)}\big)\big)({u_{k+1}}^{\alpha_{k+1}} \sqcup\!\sqcup \Sigma^*)$ and decompose w as $w = xy$ where $x \in \bigcup_{q_1,\ldots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)}$ and $y \in {u_{k+1}}^{\alpha_{k+1}} \sqcup\!\sqcup \Sigma^*$ with $|y|$ being minimal. By inductive hypothesis, we know there exists $\alpha \in [l]^k$ such that $x \in R_l^{\alpha}(u_1, \ldots, u_k) \cap S_l^{\alpha}(u_1, \ldots, u_k)$ and we then prove that $xy \in R_l^{(\alpha_1,\ldots,\alpha_{k+1})}(u_1, \ldots, u_{k+1}) \cap S_l^{(\alpha_1,\ldots,\alpha_{k+1})}(u_1, \ldots, u_{k+1})$ by distinguishing between the case in which $\alpha_{k+1} = l$ and the case in which $\alpha_{k+1} < l$. The first one is easy to handle, the second one is much trickier.

We now spell out the details.

- Suppose we have

$$\begin{aligned} w \in & \bigcup_{q_1,\ldots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)} L^{(l)}_{(u_{k+1},l)} \\ = & \Big(\bigcup_{q_1,\ldots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)} \Big) ({u_{k+1}}^{l} \sqcup\!\sqcup \Sigma^*) \ . \end{aligned}$$

  Then w can be decomposed as $w = xy$ where $x \in \bigcup_{q_1,\ldots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)}$ and $y \in {u_{k+1}}^{l} \sqcup\!\sqcup \Sigma^*$ with $|y|$ being minimal. So by inductive hypothesis, there exists $\alpha \in [l]^k$ such that $x \in R_l^{\alpha}(u_1, \ldots, u_k) \cap S_l^{\alpha}(u_1, \ldots, u_k)$. Observe that this means we have $w \in ({u_1}^{\alpha_1} \cdots {u_k}^{\alpha_k} {u_{k+1}}^{l}) \sqcup\!\sqcup \Sigma^*$ and for each $i \in [k], \alpha_i < l$, that $w \notin ({u_1}^{\alpha_1} \cdots {u_i}^{\alpha_i+1} \cdots {u_k}^{\alpha_k} {u_{k+1}}^{l}) \sqcup\!\sqcup \Sigma^*$, otherwise it would mean that $x \in ({u_1}^{\alpha_1} \cdots {u_i}^{\alpha_i+1} \cdots {u_k}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*$ by minimality

of $|y|$. Similarly, for all $i \in [k], \alpha_i < l$, it is obvious that we have

$$w = xy \in \big((u_1{}^{\alpha_1} \cdots u_{i-1}{}^{\alpha_{i-1}}) \sqcup\!\sqcup \Sigma^*\big) u_i \big((u_{i+1}{}^{\alpha_{i+1}} \cdots u_k{}^{\alpha_k} u_{k+1}{}^{l}) \sqcup\!\sqcup \Sigma^*\big)$$

as $x \in \big((u_1{}^{\alpha_1} \cdots u_{i-1}{}^{\alpha_{i-1}}) \sqcup\!\sqcup \Sigma^*\big) u_i \big((u_{i+1}{}^{\alpha_{i+1}} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big)$ and $y \in u_{k+1}{}^{l} \sqcup\!\sqcup \Sigma^*$. Hence, $w \in R_l^{(\alpha_1,\ldots,\alpha_{k+1})}(u_1,\ldots,u_{k+1}) \cap S_l^{(\alpha_1,\ldots,\alpha_{k+1})}(u_1,\ldots,u_{k+1})$.

- Or we have

$$\begin{aligned} w \notin & \bigcup_{q_1,\ldots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)} L^{(l)}_{(u_{k+1},l)} \\ =& \Big(\bigcup_{q_1,\ldots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)} \Big) (u_{k+1}{}^{l} \sqcup\!\sqcup \Sigma^*) \end{aligned}$$

but

$$w \in \bigcup_{q_1,\ldots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)} L^{(l)}_{(u_{k+1},1)} \ .$$

Let $\alpha_{k+1} \in [l-1]$ be the biggest integer in $[l-1]$ such that

$$w \in \Big(\bigcup_{q_1,\ldots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)} \Big) (u_{k+1}{}^{\alpha_{k+1}} \sqcup\!\sqcup \Sigma^*)$$

which does exist by hypothesis. We can decompose w as $w = xy$ where $x \in \bigcup_{q_1,\ldots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)}$ and $y \in u_{k+1}{}^{\alpha_{k+1}} \sqcup\!\sqcup \Sigma^*$ with $|y|$ being minimal. So by inductive hypothesis, there exists $\alpha \in [l]^k$ such that $x \in R_l^{\alpha}(u_1,\ldots,u_k) \cap S_l^{\alpha}(u_1,\ldots,u_k)$. We are now going to prove that

$$w = xy \in R_l^{(\alpha_1,\ldots,\alpha_{k+1})}(u_1,\ldots,u_{k+1}) \cap S_l^{(\alpha_1,\ldots,\alpha_{k+1})}(u_1,\ldots,u_{k+1}) \ .$$

Among the obvious things to observe is that we have $w \in (u_1{}^{\alpha_1} \cdots u_k{}^{\alpha_k} u_{k+1}{}^{\alpha_{k+1}}) \sqcup\!\sqcup \Sigma^*$ and for each $i \in [k], \alpha_i < l$, that

$$w \notin (u_1{}^{\alpha_1} \cdots u_i{}^{\alpha_i+1} \cdots u_k{}^{\alpha_k} u_{k+1}{}^{\alpha_{k+1}}) \sqcup\!\sqcup \Sigma^* \ ,$$

otherwise it would mean that $x \in (u_1{}^{\alpha_1} \cdots u_i{}^{\alpha_i+1} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*$ by minimality of $|y|$. Similarly, for all $i \in [k], \alpha_i < l$, it is obvious that we have

$$w = xy \in \big((u_1{}^{\alpha_1} \cdots u_{i-1}{}^{\alpha_{i-1}}) \sqcup\!\sqcup \Sigma^*\big) u_i \big((u_{i+1}{}^{\alpha_{i+1}} \cdots u_k{}^{\alpha_k} u_{k+1}{}^{\alpha_{k+1}}) \sqcup\!\sqcup \Sigma^*\big)$$

because $x \in \big((u_1{}^{\alpha_1} \cdots u_{i-1}{}^{\alpha_{i-1}}) \sqcup\!\sqcup \Sigma^*\big) u_i \big((u_{i+1}{}^{\alpha_{i+1}} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big)$ and $y \in u_{k+1}{}^{\alpha_{k+1}} \sqcup\!\sqcup \Sigma^*$.

Now let us show that we have $y \in \Sigma^* u_{k+1} \Sigma^*$. Assume it weren't the case: the letters in u_{k+1} are pairwise distinct and moreover y starts with the first letter of u_{k+1} by minimality of $|y|$, so u_{k+1} cannot appear as a factor of xy partly in x and partly in y and, additionally,

$$\begin{aligned} w \in & \bigcup_{q_1,\ldots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)} L^{(l)}_{(u_{k+1},1)} \\ =& \Big(\bigcup_{q_1,\ldots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)} \Big) \Sigma^* u_{k+1} \Sigma^* \ , \end{aligned}$$

so we would have $x \in (\bigcup_{q_1,\dots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)})\Sigma^* u_{k+1}\Sigma^*$. But this either contradicts the maximality of α_{k+1} or the fact that

$$w \notin \Big(\bigcup_{q_1,\dots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)} \Big)({u_{k+1}}^l \sqcup\!\sqcup \Sigma^*) \ .$$

Thus, we have $w = xy \in \big((u_1{}^{\alpha_1} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big)u_{k+1}\Sigma^*$ as $x \in (u_1{}^{\alpha_1} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*$.

Let us finish with the trickiest part, namely showing that $w \notin (u_1{}^{\alpha_1} \cdots u_k{}^{\alpha_k} u_{k+1}{}^{\alpha_{k+1}+1}) \sqcup\!\sqcup \Sigma^*$. Assume that $w \in (u_1{}^{\alpha_1} \cdots u_k{}^{\alpha_k} u_{k+1}{}^{\alpha_{k+1}+1}) \sqcup\!\sqcup \Sigma^*$. We then have that $x \in (u_1{}^{\alpha_1} \cdots u_k{}^{\alpha_k} u_{k+1}) \sqcup\!\sqcup \Sigma^*$, otherwise it would mean that $y = y_1y_2$ with $|y_1| > 0$, with $xy_1 \in (u_1{}^{\alpha_1} \cdots u_k{}^{\alpha_k} u_{k+1}) \sqcup\!\sqcup \Sigma^*$ and $y_2 \in {u_{k+1}}^{\alpha_{k+1}} \sqcup\!\sqcup \Sigma^*$, contradicting the minimality of $|y|$. We can decompose x as $x = x_1x_2$ where $x_1 \in (u_1{}^{\alpha_1} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*$ and $x_2 \in u_{k+1} \sqcup\!\sqcup \Sigma^*$ with $|x_2|$ being minimal. We claim that, actually, $x_1 \in R^\alpha_l(u_1,\dots,u_k) \cap S^\alpha_l(u_1,\dots,u_k)$, so that by inductive hypothesis, $x_1 \in \bigcup_{q_1,\dots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)}$. But since $x_2y \in {u_{k+1}}^{\alpha_{k+1}+1} \sqcup\!\sqcup \Sigma^*$, this means that

$$w = x_1x_2y \in \Big(\bigcup_{q_1,\dots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)} \Big)({u_{k+1}}^{\alpha_{k+1}+1} \sqcup\!\sqcup \Sigma^*) \ ,$$

contradicting the maximality of α_{k+1} or the fact that

$$w \notin \Big(\bigcup_{q_1,\dots,q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)} \Big)({u_{k+1}}^l \sqcup\!\sqcup \Sigma^*) \ .$$

So we can conclude that $w \notin (u_1{}^{\alpha_1} \cdots u_k{}^{\alpha_k} u_{k+1}{}^{\alpha_{k+1}+1}) \sqcup\!\sqcup \Sigma^*$.

The claim that $x_1 \in R^\alpha_l(u_1,\dots,u_k) \cap S^\alpha_l(u_1,\dots,u_k)$ remains to be shown. We directly see that $x_1 \notin (u_1{}^{\alpha_1} \cdots u_i{}^{\alpha_i+1} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*$ for all $i \in [k], \alpha_i < l$, otherwise it would mean that $x \in (u_1{}^{\alpha_1} \cdots u_i{}^{\alpha_i+1} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*$. Let now $i \in [k], \alpha_i < l$, and assume that $x_1 \notin \big((u_1{}^{\alpha_1} \cdots u_{i-1}{}^{\alpha_{i-1}}) \sqcup\!\sqcup \Sigma^*\big)u_i \big((u_{i+1}{}^{\alpha_{i+1}} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big)$. We can decompose x_1 as $x_1 = x_{1,1}x_{1,2}$ where $x_{1,1} \in (u_1{}^{\alpha_1} \cdots u_i{}^{\alpha_i}) \sqcup\!\sqcup \Sigma^*$ and $x_{1,2} \in (u_{i+1}{}^{\alpha_{i+1}} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*$ with $|x_{1,1}|$ being minimal. By hypothesis, we have $x_{1,1} \notin \big((u_1{}^{\alpha_1} \cdots u_{i-1}{}^{\alpha_{i-1}}) \sqcup\!\sqcup \Sigma^*\big)u_i\Sigma^*$, otherwise we would have

$$x_1 = x_{1,1}x_{1,2} \in \big((u_1{}^{\alpha_1} \cdots u_{i-1}{}^{\alpha_{i-1}}) \sqcup\!\sqcup \Sigma^*\big)u_i\big((u_{i+1}{}^{\alpha_{i+1}} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big) \ .$$

As previously, the letters in u_i are pairwise distinct, and $x_{1,1}$ ends with the last letter of u_i by minimality of $|x_{1,1}|$, so u_i cannot appear as a factor of x partly in $x_{1,1}$ and partly in $x_{1,2}x_2$. Thus, we have that

$$x_{1,2}x_2 \in \Sigma^* u_i\big((u_{i+1}{}^{\alpha_{i+1}} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big)$$

because we know that $x \in \big((u_1{}^{\alpha_1} \cdots u_{i-1}{}^{\alpha_{i-1}}) \sqcup\!\sqcup \Sigma^*\big)u_i\big((u_{i+1}{}^{\alpha_{i+1}} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big)$. But this means that $x = x_{1,1}x_{1,2}x_2 \in (u_1{}^{\alpha_1} \cdots u_i{}^{\alpha_i+1} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*$, a contradiction. Hence, we can deduce that for all $i \in [k], \alpha_i < l$, we have $x_1 \in \big((u_1{}^{\alpha_1} \cdots u_{i-1}{}^{\alpha_{i-1}}) \sqcup\!\sqcup \Sigma^*\big)u_i\big((u_{i+1}{}^{\alpha_{i+1}} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big)$. This finishes to show that

$$x_1 \in R^\alpha_l(u_1,\dots,u_k) \cap S^\alpha_l(u_1,\dots,u_k) \ .$$

Putting all together, we indeed also have that

$$w \in R_l^{(\alpha_1,\dots,\alpha_{k+1})}(u_1,\dots,u_{k+1}) \cap S_l^{(\alpha_1,\dots,\alpha_{k+1})}(u_1,\dots,u_{k+1})$$

in the present case.

In conclusion, in both cases,

$$w \in \bigcup_{\alpha\in[l]^{k+1}} \big(R_l^{\alpha}(u_1,\dots,u_{k+1}) \cap S_l^{\alpha}(u_1,\dots,u_{k+1})\big) \ .$$

So we can finally conclude that

$$\begin{aligned} &\bigcup_{q_1,\dots,q_{k+1}\in\{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_{k+1},q_{k+1})} \\ &= \bigcup_{\alpha\in[l]^{k+1}} \big(R_l^{\alpha}(u_1,\dots,u_{k+1}) \cap S_l^{\alpha}(u_1,\dots,u_{k+1})\big) \ . \end{aligned}$$

This concludes the proof of the lemma. □

B.3 Proof of Lemma 4.10

Before proving Lemma 4.10, we need a useful decomposition lemma, that is straightforward to prove.

Lemma B.1. *Let Σ be an alphabet and $u \in \Sigma^+$. Then, for all $\alpha \in \mathbb{N}_{>0}$, each $w \in u^\alpha \sqcup\!\sqcup \Sigma^* \cap (u^{\alpha+1} \sqcup\!\sqcup \Sigma^*)^{\complement}$ verifies*

$$w = (\prod_{i=1}^{\alpha}\prod_{j=1}^{|u|}(v_{i,j}u_j))y$$

where $v_{i,j} \in (\Sigma \setminus \{u_j\})^*$ for all $i \in [\alpha]$ and $j \in [|u|]$, and $y \in \bigcup_{i=1}^{|u|}\Big(\prod_{j=1}^{i-1}\big((\Sigma \setminus \{u_j\})^*u_j\big)(\Sigma \setminus \{u_i\})^*\Big)$.

Proof. Let Σ be an alphabet and $u \in \Sigma^+$.

Take $\alpha \in \mathbb{N}_{>0}$ and $w \in u^\alpha \sqcup\!\sqcup \Sigma^* \cap (u^{\alpha+1} \sqcup\!\sqcup \Sigma^*)^{\complement}$.

As $w \in u^\alpha \sqcup\!\sqcup \Sigma^*$, the word w can be decomposed as $w = xy$ where $x \in u^\alpha \sqcup\!\sqcup \Sigma^*$ and $|x|$ is minimal. Then, it is clearly necessarily the case that $x = \prod_{i=1}^{\alpha}\prod_{j=1}^{|u|}(v_{i,j}u_j)$ with $v_{i,j} \in (\Sigma \setminus \{u_j\})^*$ for all $i \in [\alpha]$ and $j \in [|u|]$. Moreover, as $xy \notin u^{\alpha+1} \sqcup\!\sqcup \Sigma^*$, we necessarily have that $y \notin u \sqcup\!\sqcup \Sigma^*$, so that there exists some $i \in [|u|]$ verifying that $u_1 \cdots u_{i-1}$ is a subword of y but not $u_1 \cdots u_i$. Thus, we have that $y \in \prod_{j=1}^{i-1}\big((\Sigma \setminus \{u_j\})^*u_j\big)(\Sigma \setminus \{u_i\})^*$.

This concludes the proof. □

We can now prove Lemma 4.10.

Proof of Lemma 4.10. Let Σ be an alphabet and $u \in \Sigma^+$ such that its letters are all distinct. Let $\alpha \in \mathbb{N}_{>0}$ and $x_1, x_2 \in \Sigma^*$. We let

$$L = (x_1u^\alpha x_2) \sqcup\!\sqcup \Sigma^* \cap \big((x_1u^{\alpha+1}x_2) \sqcup\!\sqcup \Sigma^*\big)^{\complement} \cap (x_1 \sqcup\!\sqcup \Sigma^*)u(x_2 \sqcup\!\sqcup \Sigma^*) \ .$$

If $|u| = 1$, the lemma follows trivially because L is piecewise testable and hence belongs to $\mathcal{L}(\mathbf{J})$, so we assume $|u| > 1$.

For each letter $a \in \Sigma$, we shall use $2|u| - 1$ distinct decorated letters of the form $a^{(i)}$ for some $i \in [\![0, 2|u| - 2]\!]$, using the convention that $a^{(0)} = a$; of course, for two distinct letters $a, b \in \Sigma$, we have that $a^{(i)}$ and $b^{(j)}$ are distinct for all $i, j \in [\![0, 2|u| - 2]\!]$. We denote by A the alphabet of these decorated letters.

For each $i \in [\![0, 2|u| - 2]\!]$, let

$$\begin{aligned} f^{(i)} \colon \Sigma &\to A \\ a &\mapsto a^{(i)} \end{aligned} .$$

For all $i \in \mathbb{N}, i \geq |u|$, we define

$$\Phi_i = (i, f^{(0)}) \prod_{j=1}^{|u|-1} (i - j, f^{(j)}) \prod_{j=2}^{|u|} (i - |u| + j, f^{(|u|+j-2)}) .$$

For all $n \in \mathbb{N}, n < |u|$, we define $\Psi_n = \varepsilon$. For all $n \in \mathbb{N}, n \geq |u|$, we define

$$\Psi_n = \prod_{i=1}^{|u|-1} (i, f^{(0)}) \prod_{i=|u|}^{n} \Phi_i .$$

Finally, let K be the language of words over A having

$$\zeta_\beta = x_1 u^{\beta-1} u \prod_{j=1}^{|u|-1} u_{|u|-j}^{(j)} \prod_{j=2}^{|u|} u_j^{(|u|+j-2)} u^{\alpha-\beta} x_2$$

for some $\beta \in [\alpha]$ as a subword but not $x_1 u^{\alpha+1} x_2$.

Claim B.2. *The sequence $(\Psi_n)_{n\in\mathbb{N}}$ of A-programs is a program-reduction from L to K.*

Let

$$\begin{aligned} s \colon \mathbb{N} &\to \mathbb{N} \\ n &\mapsto \begin{cases} 0 & \text{if } n < |u| \\ |u| - 1 + (n - |u| + 1) \cdot (2|u| - 1) & \text{otherwise} . \end{cases} \end{aligned}$$

It is direct to see that $s(n) = |\Psi_n| \leq (2|u| - 1) \cdot n$ for all $n \in \mathbb{N}$.

Therefore, using this claim, $(\Psi_n)_{n\in\mathbb{N}}$ is a program-reduction from L to K of length $s(n)$, so since K is piecewise testable and hence is recognised (classically) by some monoid from $\mathbf{J}$, Proposition 2.2 tells us that $L \in \mathcal{P}(\mathbf{J}, s(n)) = \mathcal{P}(\mathbf{J}, n)$.

Proof of claim. Let $n \in \mathbb{N}$. If $n < |u|$, then it is obvious that for all $w \in \Sigma^n$, we have $w \notin (x_1 \sqcup\!\sqcup \Sigma^*) u (x_2 \sqcup\!\sqcup \Sigma^*)$ so $w \notin L^{=n}$ and also $\Psi_n(w) = \varepsilon \notin K^{=s(n)}$, hence $L^{=n} = \emptyset = \Psi_n^{-1}(K^{=s(n)})$. Otherwise, $n \geq |u|$. We are going to show that $L^{=n} = \Psi_n^{-1}(K^{=s(n)})$.

Left-to-right inclusion. Let $w \in L^{=n}$. We want to show that $\Psi_n(w) \in K^{=s(n)}$.

We are first going to show that there exists some $\beta \in [\alpha]$ such that ζ_β is a subword of $\Psi_n(w)$. The fact that $w \in L^{=n}$ means in particular that $w \in (x_1 \sqcup\!\sqcup \Sigma^*)u(x_2 \sqcup\!\sqcup \Sigma^*)$ and we can hence decompose w as $w = y_1 z y_2$ where $y_1 \in (x_1 \sqcup\!\sqcup \Sigma^*)$ and $y_2 \in (x_2 \sqcup\!\sqcup \Sigma^*)$ with $|y_1|$ and $|y_2|$ being minimal. It follows necessarily that $z \in u^\alpha \sqcup\!\sqcup \Sigma^* \cap (u^{\alpha+1} \sqcup\!\sqcup \Sigma^*)^{\complement} \cap \Sigma^* u \Sigma^*$ by minimality of $|y_1|$ and $|y_2|$. By Lemma B.1, we have $z = \big(\prod_{i=1}^{\alpha}\prod_{j=1}^{|u|}(v_{i,j}u_j)\big)y$ where $v_{i,j} \in (\Sigma\setminus\{u_j\})^*$ for all $i \in [\alpha]$ and $j \in [|u|]$, and $y \in \bigcup_{i=1}^{|u|}\Big(\prod_{j=1}^{i-1}\big((\Sigma\setminus\{u_j\})^*u_j\big)(\Sigma\setminus\{u_i\})^*\Big)$. We know the letters in u are all distinct, so this means that there is no $\beta \in [\alpha-1]$ such that u is a factor of z partly in $\prod_{j=1}^{|u|}(v_{\beta,j}u_j)$ and partly in $\prod_{j=1}^{|u|}(v_{\beta+1,j}u_j)$, and that u cannot appear as a factor of z partly in $\prod_{j=1}^{|u|}(v_{\alpha,j}u_j)$ and partly in y either. Hence, since $z \in \Sigma^* u \Sigma^*$, by the way we decomposed z, there necessarily exists $\beta \in [\alpha]$ such that $\prod_{j=1}^{|u|}(v_{\beta,j}u_j) \in \Sigma^* u \Sigma^*$. Let $\gamma, \delta \in [n]$ such that $w_\gamma \cdots w_\delta = \prod_{j=1}^{|u|}(v_{\beta,j}u_j)$, $w_1 \cdots w_{\gamma-1} = y_1\big(\prod_{i=1}^{\beta-1}\prod_{j=1}^{|u|}(v_{i,j}u_j)\big)$ and $w_{\delta+1}\cdots w_n = \big(\prod_{i=\beta+1}^{\alpha}\prod_{j=1}^{|u|}(v_{i,j}u_j)\big)yy_2$. By the way β is defined, we have $w_{\delta-|u|+1}\cdots w_\delta = u$, because δ is the first and only position in w with the letter $u_{|u|}$ within the interval $[\![\gamma, \delta]\!]$ verifying that $w_\gamma \cdots w_{\delta-1}$ contains $u_1 \cdots u_{|u|-1}$ as a subword, and we observe additionally that $\delta \geq \gamma + |u| - 1 \geq |u|$. This means that

$$\begin{aligned}
&\Phi_\delta(w)\\
=&f^{(0)}(w_\delta)f^{(1)}(w_{\delta-1})\cdots f^{(|u|-1)}(w_{\delta-|u|+1})f^{(|u|)}(w_{\delta-|u|+2})\cdots f^{(2|u|-2)}(w_\delta)\\
=&u_{|u|}\prod_{j=1}^{|u|-1}u_{|u|-j}^{(j)}\prod_{j=2}^{|u|}u_j^{(|u|+j-2)}\ .
\end{aligned}$$

Moreover,

$$\prod_{i=1}^{\gamma-1}f^{(0)}(w_i) = w_1\cdots w_{\gamma-1} = y_1\big(\prod_{i=1}^{\beta-1}\prod_{j=1}^{|u|}(v_{i,j}u_j)\big)\ ,$$

$$\prod_{i=\delta-|u|+1}^{\delta-1}f^{(0)}(w_i) = w_{\delta-|u|+1}\cdots w_{\delta-1} = u_1\cdots u_{|u|-1}$$

and

$$\prod_{i=\delta+1}^{n}f^{(0)}(w_i) = w_{\delta+1}\cdots w_n = \big(\prod_{i=\beta+1}^{\alpha}\prod_{j=1}^{|u|}(v_{i,j}u_j)\big)yy_2\ .$$

So as $\prod_{i=1}^{\gamma-1}(i, f^{(0)})\prod_{i=\delta-|u|+1}^{\delta-1}(i, f^{(0)})\Phi_\delta\prod_{i=\delta+1}^{n}(i, f^{(0)})$ is a subword of Ψ_n, we have that

$$\zeta_\beta = x_1 u^{\beta-1} u \prod_{j=1}^{|u|-1}u_{|u|-j}^{(j)}\prod_{j=2}^{|u|}u_j^{(|u|+j-2)}u^{\alpha-\beta}x_2$$

is a subword of $\Psi_n(w)$.

We secondly show that $x_1 u^{\alpha+1} x_2$ cannot be a subword of $\Psi_n(w)$. But this is direct by construction of Ψ_n, otherwise we would have that $x_1 u^{\alpha+1} x_2$ is a subword of w, contradicting the fact that $w \in L^{=n}$.

Hence, $\Psi_n(w) \in K^{=s(n)}$, and since this is true for all $w \in L^{=n}$, we have $L^{=n} \subseteq \Psi_n^{-1}(K^{=s(n)})$.

Right-to-left inclusion. We are going to prove the "contrapositive inclusion".

Let $w \in \Sigma^n \setminus L^{=n}$. We want to show that $\Psi_n(w) \notin K^{=s(n)}$.

Let us start with the easy cases. If we have $w \notin (x_1 u^\alpha x_2) \sqcup\!\sqcup \Sigma^*$, then it means that $x_1 u^\alpha x_2$ is not a subword of w and hence, by construction of Ψ_n, not a subword of $\Psi_n(w)$ either, so that there does not exist any $\beta \in [\alpha]$ such that ζ_β is a subword of $\Psi_n(w)$. Similarly, if we have $w \in (x_1 u^{\alpha+1} x_2) \sqcup\!\sqcup \Sigma^*$, then it means that $x_1 u^{\alpha+1} x_2$ is a subword of w and hence, by construction of Ψ_n, a subword of $\Psi_n(w)$.

We now assume that $w \in (x_1 u^\alpha x_2) \sqcup\!\sqcup \Sigma^* \cap \big((x_1 u^{\alpha+1} x_2) \sqcup\!\sqcup \Sigma^*\big)^{\complement}$ while $w \notin (x_1 \sqcup\!\sqcup \Sigma^*) u (x_2 \sqcup\!\sqcup \Sigma^*)$. We want to show that in this case, there does not exist any $\beta \in [\alpha]$ such that ζ_β is a subword of $\Psi_n(w)$. Suppose for a contradiction that such a β exists; our goal is to show, through a careful observation of what this implies on the letters in w by examining how Ψ_n decorates the letters, that this contradictingly entails $x_1 u^{\alpha+1} x_2$ is a subword of w.

Since ζ_β is a subword of $\Psi_n(w)$, it is not too difficult to see there exist

$$p_1, \dots, p_{|x_1|+(\beta-1)\cdot|u|}, q_1, \dots, q_{3|u|-2}, r_1, \dots, r_{(\alpha-\beta)\cdot|u|+|x_2|} \in [n]$$

verifying that

$$w_{p_1} \cdots w_{p_{|x_1|+(\beta-1)\cdot|u|}} = x_1 u^{\beta-1} \ ,$$

$$w_{q_1} \cdots w_{q_{3|u|-2}} = u \prod_{j=1}^{|u|-1} u_{|u|-j} \prod_{j=2}^{|u|} u_j \ ,$$

$$w_{r_1} \cdots w_{r_{(\alpha-\beta)\cdot|u|+|x_2|}} = u^{\alpha-\beta} x_2$$

and

$$\begin{aligned} &(p_1, f^{(0)}) \cdots (p_{|x_1|+(\beta-1)\cdot|u|}, f^{(0)}) (q_1, f^{(0)}) \cdots (q_{|u|}, f^{(0)}) \\ &(q_{|u|+1}, f^{(1)}) \cdots (q_{2|u|-1}, f^{(|u|-1)}) (q_{2|u|}, f^{(|u|)}) \cdots (q_{3|u|-2}, f^{(2|u|-2)}) \\ &(r_1, f^{(0)}) \cdots (r_{(\alpha-\beta)\cdot|u|+|x_2|}, f^{(0)}) \end{aligned}$$

is a subword of Ψ_n. By construction of Ψ_n, we have

$$p_1 < \cdots < p_{|x_1|+(\beta-1)\cdot|u|} < q_1 < \cdots < q_{|u|} < r_1 < \cdots < r_{(\alpha-\beta)\cdot|u|+|x_2|} \ ,$$

so this implies that w can be decomposed as $w = y_1 z y_2$ where $y_1 \in x_1 \sqcup\!\sqcup \Sigma^*$, where $z \in u^\alpha \sqcup\!\sqcup \Sigma^*$ and $y_2 \in x_2 \sqcup\!\sqcup \Sigma^*$, the positions $p_1, \dots, p_{|x_1|}$ corresponding to letters in y_1, the positions $p_{|x_1|+1}, \dots, p_{|x_1|+(\beta-1)\cdot|u|}, q_1, \dots, q_{|u|}$, $r_1, \dots, r_{(\alpha-\beta)\cdot|u|}$ corresponding to letters in z and the positions $r_{(\alpha-\beta)\cdot|u|+1}, \dots$, $r_{(\alpha-\beta)\cdot|u|+|x_2|}$ corresponding to letters in y_2.

We are now going to show that, in fact, $q_{|u|} < q_{2|u|-1} < q_{2|u|} < \cdots < q_{3|u|-2} < r_1$, which implies $z \in u^{\alpha+1} \sqcup\!\sqcup \Sigma^*$ and thus the contradiction we are aiming for. Since $w \notin (x_1 \sqcup\!\sqcup \Sigma^*) u (x_2 \sqcup\!\sqcup \Sigma^*)$, we have $z \notin \Sigma^* u \Sigma^*$, hence as $w_{q_{|u|}} = u_{|u|}$ and $|u| > 1$, there must exist $j \in [|u|-1]$ such that $w_{q_{|u|}-j} \neq u_{|u|-j}$ and $w_{q_{|u|}-\iota} = u_{|u|-\iota}$ for all $\iota \in [\![0, j-1]\!]$. By construction of Ψ_n, we know that $q_{|u|+j} \geq q_{|u|} - j$ (because the instructions with $f^{(j)}$ after an instruction with

$f^{(0)}$ querying position $p \in [n]$ all query a position at least equal to $p-j$), but since $u_{|u|-j} \neq w_{q_{|u|}-j}$ and $u_{|u|-j} \neq u_{|u|-\iota} = w_{q_{|u|}-\iota}$ for all $\iota \in [\![0, j-1]\!]$ as the letters in u are all distinct, we get that $q_{|u|+j} > q_{|u|}$. By (backward) induction, we can show that for all $\iota \in [\![j+1, |u|-1]\!]$, we have $q_{|u|+\iota} > q_{|u|}$. Indeed, given $\iota \in [\![j+1, |u|-1]\!]$, we have $q_{|u|+\iota-1} > q_{|u|}$, either by inductive hypothesis or directly in the base case $\iota = j+1$ by what we have just seen. So by construction of Ψ_n, we know that $q_{|u|+\iota} \geq q_{|u|}$ (because the instructions with $f^{(\iota)}$ after an instruction with $f^{(\iota-1)}$ querying position $p \in [n]$ all query a position at least equal to $p-1$), but since $u_{|u|-\iota} \neq u_{|u|} = w_{q_{|u|}}$ as the letters in u are all distinct, it follows that $q_{|u|+\iota} > q_{|u|}$. Therefore, we have that $q_{2|u|-1} > q_{|u|}$. Moreover, by construction of Ψ_n, we also have $q_{2|u|-1} < q_{2|u|} < \cdots < q_{3|u|-2} < r_1$ (because for each $\iota \in [\![0, |u|-2]\!]$, the instructions with $f^{(|u|+\iota)}$ after an instruction with $f^{(|u|+\iota-1)}$ querying position $p \in [n]$ all query a position at least equal to $p+1$ and similarly for the instructions with $f^{(0)}$ after an instruction with $f^{(2|u|-2)}$). So, to conclude, we have $p_1 < \cdots < p_{|x_1|+(\beta-1)\cdot|u|} < q_1 < \cdots < q_{|u|} < q_{2|u|-1} < q_{2|u|} < \cdots < q_{3|u|-2} < r_1 < \cdots < r_{(\alpha-\beta)\cdot|u|+|x_2|}$ and

$$\begin{aligned}&w_{p_1} \cdots w_{p_{|x_1|+(\beta-1)\cdot|u|}} w_{q_1} \cdots w_{q_{|u|}} w_{q_{2|u|-1}} w_{q_{2|u|}} \cdots w_{q_{3|u|-2}} w_{r_1} \cdots w_{r_{(\alpha-\beta)\cdot|u|+|x_2|}} \\ =&x_1 u^{\beta-1} u u_1 u_2 \cdots u_{|u|} u^{\alpha-\beta} x_2 = x_1 u^{\alpha+1} x_2 \ .\end{aligned}$$

This implies that $w \in (x_1 u^{\alpha+1} x_2) \sqcup\!\sqcup \Sigma^*$, a contradiction. So there does not exist $\beta \in [\alpha]$ such that ζ_β is a subword of $\Psi(w)$.

Therefore, in every case $\Psi_n(w) \notin K^{=s(n)}$, and since this is true for all $w \in \Sigma^n \setminus L^{=n}$, we have $\Sigma^n \setminus L^{=n} \subseteq \Psi_n^{-1}(A^{s(n)} \setminus K^{=s(n)})$, which is equivalent to $L^{=n} \supseteq \Psi_n^{-1}(K^{=s(n)})$.

This concludes the proof of the claim. □

And the one of the lemma. □

B.4 Proof of Proposition 4.11

Proof of Proposition 4.11. Let Σ be an alphabet, $l \in \mathbb{N}_{>0}$ and $u_1, \ldots, u_k \in \Sigma^+$ ($k \in \mathbb{N}_{>0}$) such that for each $i \in [k]$, the letters in u_i are all distinct. Let $\alpha \in [l]^k$.

For each $i \in [k]$ verifying $\alpha_i < l$, we define

$$\begin{aligned}L_i =&(u_1{}^{\alpha_1} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^* \cap \big((u_1{}^{\alpha_1} \cdots u_i{}^{\alpha_i+1} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big)^{\complement} \cap \\ &\big((u_1{}^{\alpha_1} \cdots u_{i-1}{}^{\alpha_{i-1}}) \sqcup\!\sqcup \Sigma^*\big) u_i \big((u_{i+1}{}^{\alpha_{i+1}} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*\big) \ .\end{aligned}$$

It is immediate to show that

$$R_l^\alpha(u_1, \ldots, u_k) \cap S_l^\alpha(u_1, \ldots, u_k) = (u_1{}^{\alpha_1} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^* \cap \bigcap_{i \in [k], \alpha_i < l} L_i \ .$$

By Lemma 4.10, $L_i \in \mathcal{P}(\mathbf{J})$ for each $i \in [k]$ verifying $\alpha_i < l$. Moreover, since $(u_1{}^{\alpha_1} \cdots u_k{}^{\alpha_k}) \sqcup\!\sqcup \Sigma^*$ obviously is a piecewise testable language, it belongs to $\mathcal{P}(\mathbf{J})$. Thus, we can conclude that $R_l^\alpha(u_1, \ldots, u_k) \cap S_l^\alpha(u_1, \ldots, u_k)$ belongs to $\mathcal{P}(\mathbf{J})$ by closure of $\mathcal{P}(\mathbf{J}) \cap \mathcal{R}\mathsf{eg}$ under intersection, Proposition 2.1. □

B.5 Proof of Proposition 4.13

Proof of Proposition 4.13. Let Σ be an alphabet, $l \in \mathbb{N}_{>0}$ and $u_1, \ldots, u_k \in \Sigma^+$ ($k \in \mathbb{N}_{>0}$).

Let $d = \max_{i \in [k]} |u_i|$. If $d = 1$, then the result is straightforward because the language $[u_1, \ldots, u_k]_l$ then belongs to $\mathcal{L}(\mathbf{J})$, so now we assume $d \geq 2$. We let $\Sigma_d = \Sigma \times \mathbb{Z}/d\mathbb{Z}$ and for all $w \in \Sigma^*$, for all $i \in \mathbb{Z}/d\mathbb{Z}$, we define $\widetilde{w}^i = \prod_{j=1}^{|w|} (w_j, (j + i - 1) \mod d)$. We also let $\widetilde{w} = \widetilde{w}^0$ for all $w \in \Sigma^*$.

For all $v \in \Sigma^+, |v| \leq d$, we define $\mu(v, 1) = v$ and

$$\mu(v, l) = \underbrace{v_1, \ldots, v_{|v|}, \ldots\ldots\ldots, v_1, \ldots, v_{|v|}}_{l \text{ times}} .$$

For all $v_1, \ldots, v_{k'} \in \Sigma^+$ ($k' \in \mathbb{N}_{>0}$) such that $|v_i| \leq d$ for each $i \in [k']$, we let

$$[v_1, \ldots, v_{k'}]_{l,d} = \bigcup_{i_1, \ldots, i_{k'} \in \mathbb{Z}/d\mathbb{Z}} [\widetilde{v_1}^{i_1}, \ldots, \widetilde{v_{k'}}^{i_{k'}}]_l ,$$

a language over Σ_d, that does belong to $\mathcal{P}(\mathbf{J})$ by Corollary 4.12 and closure of $\mathcal{P}(\mathbf{J}) \cap \mathcal{R}\mathsf{eg}$ under finite union (Proposition 2.1), because since $|v_i| \leq d$ for each $i \in [k']$, each $\widetilde{v_i}^j$ for $j \in \mathbb{Z}/d\mathbb{Z}$ has all distinct letters.

This implies that for all $q_1, \ldots, q_k \in \{1, l\}$, we have that $[\mu(u_1, q_1), \ldots, \mu(u_k, q_k)]_{l,d}$ does belong to $\mathcal{P}(\mathbf{J})$, so that

$$\bigcup_{q_1, \ldots, q_k \in \{1,l\}} [\mu(u_1, q_1), \ldots, \mu(u_k, q_k)]_{l,d}$$

is a language over Σ_d belonging to $\mathcal{P}(\mathbf{J})$.

Now, it is not so difficult to see that

$$\begin{aligned} [u_1, \ldots, u_k]_l &= \bigcup_{q_1, \ldots, q_k \in \{1,l\}} L^{(l)}_{(u_1,q_1)} \cdots L^{(l)}_{(u_k,q_k)} \\ &= \left\{ w \in \Sigma^* \;\middle|\; \widetilde{w} \in \bigcup_{q_1, \ldots, q_k \in \{1,l\}} [\mu(u_1, q_1), \ldots, \mu(u_k, q_k)]_{l,d} \right\} , \end{aligned}$$

which allows us to conclude that the sequence $(\Psi_n)_{n \in \mathbb{N}}$ of Σ_d-programs such that $\Psi_n(w) = \widetilde{w}$ for all $n \in \mathbb{N}$ and $w \in \Sigma^n$ is a program-reduction from $[u_1, \ldots, u_k]_l$ to $\bigcup_{q_1, \ldots, q_k \in \{1,l\}} [\mu(u_1, q_1), \ldots, \mu(u_k, q_k)]_{l,d}$ of length n. Hence, $[u_1, \ldots, u_k]_l$ does also belong to $\mathcal{P}(\mathbf{J})$ by Proposition 2.2. □